

RL-TR-96-285
Final Technical Report
April 1997



MULTIDISCIPLINARY RESEARCH ON ADVANCED, HIGH-SPEED, ADAPTIVE SIGNAL PROCESSING FOR RADAR SENSORS

University of Southern California

Sponsored by
Advanced Research Projects Agency
ARPA Order No. B432-03

DTIC QUALITY INSPECTED 2

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

19970620 035

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

Rome Laboratory
Air Force Materiel Command
Rome, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-96-285 has been reviewed and is approved for publication.

APPROVED: *William L. Melvin Jr*
WILLIAM L. MELVIN, JR., Capt, USAF
Project Engineer

FOR THE COMMANDER: *Gary D. Barmore*
GARY D. BARMORE, Major, USAF
Deputy Director
Surveillance & Photonics Directorate

If your address has changed or if you wish to be removed from the Rome Laboratory mailing list, or if the addressee is no longer employed by your organization, please notify RL/OCSS, 26 Electronic Pky, Rome, NY 13441-4514. This will assist us in maintaining a current mailing list.

Do not return copies of this report unless contractual obligations or notices on a specific document require that it be returned.

MULTIDISCIPLINARY RESEARCH ON ADVANCED,
HIGH-SPEED, ADAPTIVE SIGNAL PROCESSING FOR
RADAR SENSORS

Contractor: University of Southern California
Contract Number: F30602-95-1-0001
Effective Date of Contract: 23 January 1995
Contract Expiration Date: 30 September 1996
Program Code Number: B432-03
Short Title of Work: Multidisciplinary Research on Advanced,
High-Speed, Adaptive Signal Processing
Period of Work Covered: Jan 95 - Sep 96

Principal Investigator: Irving Reed
Phone: (213) 740-7335
RL Project Engineer: William L. Melvin, Capt, USAF
Phone: (315) 330-1896

Approved for Public Release, Distribution Unlimited

This research was supported by the Advanced Research Projects
Agency of the Department of Defense and was monitored by
Capt William L. Melvin, RL/OCSS, 26 Electronic Pky, Rome, NY.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
<small>Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.</small>				
1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE April 1997	3. REPORT TYPE AND DATES COVERED Final Jan 95 - Sep 96		
4. TITLE AND SUBTITLE MULTIDISCIPLINARY RESEARCH ON ADVANCED, HIGH-SPEED, ADAPTIVE SIGNAL PROCESSING FOR RADAR SENSORS		5. FUNDING NUMBERS C - F30602-95-1-0001 PE - 63226E PR - MSEF TA - 05 WU - 03		
6. AUTHOR(S) Irving Reed, Chrysostomos L. Nikias, and Viktor K. Prasanna				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) University of Southern California Department of Electrical Engineering University Park Los Angeles, CA 90089-2564		8. PERFORMING ORGANIZATION REPORT NUMBER N/A		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Advanced Research Projects Agency 3701 North Fairfax Drive Arlington, VA 22203-1714		10. SPONSORING/MONITORING AGENCY REPORT NUMBER RL-TR-96-285		
11. SUPPLEMENTARY NOTES Rome Laboratory Project Engineer: William L. Melvin, Capt, USAF/OCSS/(315) 330-1896				
12a. DISTRIBUTION AVAILABILITY STATEMENT Approved for Public Release, Distribution Unlimited		12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words) This report addresses two major components of research for high-speed, space-time adaptive processing (STAP) for radar sensors, namely (i) the development of advanced algorithms for detection and parameter estimation of weak targets in the presence of jamming and clutter, and (ii) the mapping of the algorithms onto massively parallel computing architectures for high-speed implementation. First, advances in detection and estimation for STAP applications are achieved using joint Gaussian statistics. A cross-spectral method, an optimal technique for reduced-rank STAP, and a simultaneous CFAR detection and maximum likelihood estimation STAP algorithm for airborne radar is introduced. Secondly, this report discusses new methods for parameter estimation with symmetric alpha-stable distributions and fractional lower-order moments. A Cauchy beamformer is proposed, along with a new joint spatial- and Doppler-frequency, high resolution estimation technique based on eigen-decomposition of the covariance matrix. Finally, this report investigates the issue of mapping the above signal processing algorithms to scaleable, portable, parallel implementations.				
14. SUBJECT TERMS Space-Time Adaptive Processing, Cross-spectral reduced-rank, STAP, CFAR detection, alphastable processes, lower-order moments, parallel processing		15. NUMBER OF PAGES 158		
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED		18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED		16. PRICE CODE
19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED		20. LIMITATION OF ABSTRACT UL		

Accomplishments

In this grant we addressed issues related to the two major components of our proposed research for Space-Time Adaptive Processing (STAP) applications, namely (i) the development of advanced algorithms for detection and parameter estimation of weak targets in the presence of jamming and radar clutter, and (ii) the mapping of the algorithms into massively parallel architectures for their high-speed implementation. Our accomplishments during this grant may be summarized as follows:

Advances in Detection and Estimation for STAP Applications with Joint Gaussian Statistics

Investigator: Professor Irving S. Reed

1. The complete theoretical development of the *Cross-Spectral Method*, an optimal technique for reduced-rank STAP. The performance measure for optimality is maximizing the output signal-to-interference plus noise ratio, which is equivalent to minimizing the mean-square error of the processor. A proof of optimality has been obtained and, while counter-intuitive, the resulting solution is shown to be different from that yielded by the principal component approximation of the array covariance matrix. The rank of the resulting weight vector is reduced by the cross-spectral method, in contrast to the principal component inverse (PCI)/Eigencanceler methods. Finally, a low complexity filtering-based implementation of the cross-spectral method has been investigated;
2. The design of a simultaneous CFAR detection and maximum likelihood (ML) estimation STAP algorithm for airborne radar. Generalization of our previous results on element-space post-Doppler STAP to fully adaptive, beam-space post-Doppler processors. This study demonstrates that the finer resolution leads to a reduction in the detection capability. This implies that one may trade-off the accuracy of ML estimation with the performance of the CFAR detection criterion;
3. The development of a fast constant false alarm (CFAR) detection algorithm for STAP. This approach is based on the eigen-decomposition of the space-time radar covariance matrix. The method utilized by this algorithm does not require a matrix inversion, thereby reducing the computational complexity requirements of the processor.

Robust Detection and Estimation with Alpha-Stable Distributions for STAP Applications

Investigator: Professor Chrysostomos L. Nikias

1. The characterization and development of methods for parameter estimation of symmetric alpha-stable ($S\alpha S$) distributions by means of fractional lower- and negative-order moments. The modeling of the amplitude statistics of radar clutter by means of $S\alpha S$ distributions and the estimation of the parameters of the stable distributions from real clutter of the Mountain Top Database;

2. The development of an adaptive matched-filter detector for the case of a signal embedded in heavy-tailed noise modeled as a sub-Gaussian, alpha-stable process. The implementation of a generalized likelihood ratio detector which employs robust estimates of the unknown noise underlying matrix and the unknown signal strength. The comparison of the performance of the new detector to the performance of the optimum detector under completely known signal and Gaussian noise characteristics;
3. The theoretical analysis on Cramér-Rao bounds for target angle and Doppler estimation for airborne radar operating in interference modeled as a Cauchy process. The development of optimal, maximum likelihood-based approaches to the estimation problem and the introduction of the *Cauchy Beamformer* which exhibits robust performance in a wide range of impulsive noise environments. The development of new joint spatial- and Doppler-frequency high-resolution estimation techniques based on the eigendecomposition of the *covariation matrix* of the space-time radar measurements.

Scalable Portable Parallel Algorithms for STAP Applications

Investigator: Professor Viktor K. Prasanna

1. The development of efficient algorithms for parallel Higher Order Post-Doppler Processing. Our algorithms, which exploit remapping data between various computation steps and scheduling the computations to overlap communication, lead to scalable parallel algorithms. The previous approach employs row wrap mapping scheme to avoid data remapping. However, the communication time increases as the number of processors increases. Our scalability analysis and experimental results show that the frequent short message communications of the earlier fine grain algorithms can be eliminated by data remapping;
2. The development of parallel weight computation steps of Element Space Pre-Doppler and Element Space Post-Doppler STAP, which are the major computational bottlenecks in the entire processing. The tasks are partitioned evenly. The parallelism is achieved by distributing whole tasks among the processing elements. The reason for employing task level parallelism is that the problem sizes of the tasks in Element Space STAP are too small to parallelize efficiently. Despite advances in network technology, state-of-the-art HPC platforms still consume longer time in communicating than in computing a unit data. Hence, coarse grain parallelism is more suitable for these platforms. Thus, eliminating sparse short communications by providing data required in each task based on owner compute rule, can achieve much greater speed-up;
3. The development of efficient data distribution algorithms which exploit intermediate data remapping schemes and communication contexts. The proposed scheme is a two phase approach which exploits combine-and-forward techniques. By using intermediate data mapping, the introduced algorithms effectively reduce startup overhead and node contention. Our algorithms also utilize pipelined communication scheduling.

Contents

1	Detection and Estimation with Joint Gaussian Statistics	1
1.1	The Cross-Spectral Metric and Optimal Rank Reduction	2
1.1.1	Space-Time Adaptive Processing	2
1.1.2	The GSC Form Processor	4
1.1.3	Partially Adaptive Processing	7
1.1.4	Example Using the Mountaintop Radar Parameters	9
1.1.5	Conclusions	14
1.2	A Simultaneous CFAR Detection and ML Estimation STAP Algorithm for Airborne Radar	14
1.2.1	Introduction	15
1.2.2	Formulation of the Problem	16
1.2.3	Derivation of the Test Statistic (when Noise Covariance is Known)	16
1.2.4	The Maximum Likelihood Estimator of Target Direction	19
1.2.5	Performance Analysis of the Detector (when the Noise Covariance is Unknown)	20
1.2.6	Performance Analyses of Two Partially Adaptive STAP Detectors	25
1.2.7	Conclusion	28
1.3	A Fast CFAR Detection Space-Time Adaptive Processing Algorithm	28
1.3.1	Introduction	28
1.3.2	Derivation of the Test Function	29
1.3.3	Rank Reduction of the Clutter-Plus-Noise Covariance	30
2	Detection and Estimation with Stable Distributions	44
2.1	Alpha-Stable Random Variables and Processes	45
2.1.1	The Class of Real $S\alpha S$ Distributions	45
2.1.2	Symmetric Sub-Gaussian Alpha-Stable Processes	47
2.1.3	Complex $S\alpha S$ Random Variables and Covariations	48
2.1.4	Generation of Complex Isotropic $S\alpha S$ Random Variables	49
2.2	Parameter Estimation with Fractional Lower-Order Moments	50
2.2.1	Mountaintop Data Analysis: Clutter Modeling Using $S\alpha S$ Theory	52
2.3	Signal Detection in Sub-Gaussian Impulsive Interference	53
2.3.1	Estimation of the Underlying Matrix of a Sub-Gaussian Vector	53
2.3.2	Data-Adaptive Algorithms for Coherent Signal Detection	55
2.3.3	Performance Analysis and Computer Illustration	56
2.3.4	The Special Case of a Single Observation	57

2.3.5	Summary and Conclusions	57
2.4	Target Angle and Doppler Estimation for Airborne Radar in Impulsive Interference Modeled as a Stable Process	58
2.4.1	Maximum Likelihood Estimation in Alpha-Stable Noise: The Cauchy Beam- former	58
2.4.2	Subspace-Based STAP Using Lower-Order Statistics	63
3	Scalable Portable Parallel Algorithms for STAP	98
3.1	Overview	99
3.1.1	Modeling High Performance Computing Platforms	102
3.1.2	Scalability	103
3.1.3	Portability and MPI	104
3.2	HOPD STAP and its Computational Requirements	106
3.3	Scalable Parallel Algorithms for HOPD STAP	107
3.3.1	Computational Model	107
3.3.2	Scalability and Portability	109
3.3.3	Data Remapping, Scheduling, and Overlapping Computation with Com- munication	109
3.4	Comparison with previous approaches	114
3.5	Experimental Results	117
3.6	A Methodology for the Design of Scalable Solutions	122
3.7	Computational Characteristics of Element-Space STAP	123
3.8	Scalable Portable Algorithms for ESPrD and ESPsD	129
3.9	Conclusion	130

List of Tables

2.1	Performance comparison of the $\log \text{SaS} $ Estimator versus the Sinc Function Estimator. The sample size is 5000, the true values are $\alpha = 1.5$ and $\gamma = 1$	52
2.2	GSNR and average PSNR for different values of M	61
2.3	GSNR and average PSNR for different values of γ	62
2.4	GSNR and average PSNR for different values of α	62
2.5	GSNR and average PSNR for different values of α	67
3.1	Communication features of various HPC platforms.	109
3.2	Timing Results for computing FFT and Data Remapping on T3D and SP2. . . .	120
3.3	Total Execution Times on SP-2 for $M=N=64$, $L=4096$	120

List of Figures

1.1	3-Dimensional CPI Data Cube	3
1.2	The full-rank GSC MVDR processor.	5
1.3	The full-rank GSC MVDR processor in principal coordinates.	6
1.4	The reduced-rank GSC MVDR processor.	7
1.5	The power spectrum of the data	11
1.6	The full-rank Wiener filter power spectrum	11
1.7	The reduced-rank eigen-subspace Wiener filter power spectrum	12
1.8	The reduced rank cross-spectral subspace Wiener filter power spectrum	12
1.9	The output SINR of the reduced-rank processors as a function of the rank of the data	13
1.10	Fully adaptive STAP.	35
1.11	Element-space, post-doppler STAP.	36
1.12	Beam-space, post-doppler partial STAP.	37
1.13	Probability of detection of fully adaptive STAP (M=10,N=10).	38
1.14	Probability of detection of fully adaptive STAP (M=5,N=5).	38
1.15	Probability of detection of partially adaptive STAP (M=10,N=10).	39
1.16	Probability of detection of partially adaptive STAP (M=5,N=5).	39
1.17	Comparison of the probability of detection between fully and partially adaptive STAP (M=10,N=10)	40
1.18	Comparison of the probability of detection between fully and partially adaptive STAP (M=5,N=5)	40
2.1	Standard $S\alpha S$ densities.	76
2.2	A close-up view of the tails of the densities in Figure 2.1.	76
2.3	Typical realizations of sub-Gaussian random vectors.	77
2.4	Measured I/Q-components of radar clutter: Azimuth: 15° , estimated mean: $0.0043 + j0.0099$, estimated $\alpha = 0.7318$, $\gamma = 0.0413$	78
2.5	Comparison of the empirical, $S\alpha S$, and Gaussian amplitude probability distributions.	78
2.6	Measured I/Q-components of radar clutter: Azimuth: 45° , estimated mean: $-0.0121 - j0.0247$, estimated $\alpha = 0.8301$, $\gamma = 0.0537$	79
2.7	Comparison of the empirical, $S\alpha S$, and Gaussian amplitude probability distributions.	79
2.8	Measured I/Q-components of radar clutter: Azimuth: 60° , estimated mean: $0.0051 + j0.005$, estimated $\alpha = 0.6886$, $\gamma = 0.0394$	80
2.9	Comparison of the empirical APD with Gaussian and $S\alpha S$ models.	80
2.10	Measured I/Q-components of radar clutter: Azimuth: 105° , estimated mean: $0.028 - j0.0241$, estimated $\alpha = 1.1045$, $\gamma = 0.0249$	81

2.11	Comparison of the empirical APD with Gaussian and S α S models.	81
2.12	Measured I/Q-components of radar clutter: Azimuth: 120°, estimated mean: -0.0199 + j 0.0215, estimated $\alpha = 1.1314$, $\gamma = 0.0171$	82
2.13	Comparison of the empirical APD with Gaussian and S α S models.	82
2.14	Measured I/Q-components of radar clutter: Azimuth: 240°, estimated mean: -0.0092 - j 0.007, estimated $\alpha = 0.8358$, $\gamma = 0.0337$	83
2.15	Comparison of the empirical APD with Gaussian and S α S models.	83
2.16	Measured I/Q-components of radar clutter: Azimuth: 315°, estimated mean: 0.017 + j 0.0079, estimated $\alpha = 1.0304$, $\gamma = 0.0193$	84
2.17	Comparison of the empirical APD with Gaussian and S α S models.	84
2.18	Illustration of the performance of estimators of the underlying matrix of a sub-Gaussian vector.	85
2.19	Comparison of the small sample performance of the Gaussian (dotted line) and the Cauchy (solid line) detector.	85
2.20	Performance of the Gaussian (left column) and the Cauchy (right column) detector as a function of the characteristic exponent α	86
2.21	MSE of the estimated DOA and CRB as functions of the number of snapshots M . (a) Exact signal knowledge, (b) Least-squares estimate of the signal.	87
2.22	MSE of the estimated DOA and CRB as functions of the GSNR. (a) Exact signal knowledge, (b) Least-squares estimate of the signal.	87
2.23	MSE of the estimated DOA and CRB as functions of the characteristic exponent α . (a) Exact signal knowledge, (b) Least-squares estimate of the signal.	88
2.24	MUSIC (a-b) and ROC-MUSIC (c-d) angle-Doppler spectra ($N = 5$, $M = 10$, $\Theta = [-20^\circ, -40^\circ, 40^\circ]$ $\mathbf{D} = [-0.3, -0.2, 0.3]$). Additive stable noise ($\alpha = 1.1$, $\gamma = 2$).	89
2.25	MUSIC (a-b) and ROC-MUSIC (c-d) angle-Doppler spectra ($N = 5$, $M = 10$, $\Theta = [-20^\circ, 30^\circ, 40^\circ]$ $\mathbf{D} = [-0.3, 0.2, 0.3]$). Additive stable noise ($\alpha = 1.3$, $\gamma = 2$).	90
2.26	MUSIC (a-b) and ROC-MUSIC (c-d) angle-Doppler spectra ($N = 5$, $M = 10$, $\Theta = [-20^\circ, -40^\circ, 40^\circ]$ $\mathbf{D} = [-0.3, -0.2, 0.3]$). Additive stable noise ($\alpha = 1.5$, $\gamma = 4$).	91
2.27	MUSIC (a-b) and ROC-MUSIC (c-d) angle-Doppler spectra ($N = 5$, $M = 10$, $\Theta = [-20^\circ, -40^\circ, 40^\circ]$ $\mathbf{D} = [-0.3, -0.2, 0.3]$). Additive stable noise ($\alpha = 1.8$, $\gamma = 4$).	92
2.28	MUSIC (a-b) and ROC-MUSIC (c-d) angle-Doppler spectra ($N = 5$, $M = 10$, $\Theta = [-20^\circ, -40^\circ, 40^\circ]$ $\mathbf{D} = [-0.3, -0.2, 0.3]$). Additive Gaussian noise ($\alpha = 2.0$, $\gamma = 4$).	93
2.29	Probability of resolution (a) and mean square error (b) as functions of the characteristic exponent α	94
2.30	Probability of resolution and mean square error as functions of the source angular separation, (a-b): $\alpha = 1.5$, (c-d): $\alpha = 1.8$	95
3.1	Typical HPC platform	103
3.2	Data cube and FFT & QRD computations.	108
3.3	Data remapping process.	111
3.4	Boundary data exchange.	112
3.5	The parallel QR decomposition algorithm.	115
3.6	The timing diagram of the parallel QR decomposition algorithm for the first two steps. Four processors are assumed for the sake of illustration.	116

3.7	Estimated T_{comm} and T'_{comm} on SP-2 for $M=64$, $N=64$, $L_{ls}=1024$. Each data is assumed as single precision complex number (8 bytes).	118
3.8	Asynchronous Nonblocking communication v.s. Synchronous Blocking communication for Remapping.	119
3.9	Experimental results of data remapping with overlapping for $M=64$, $N=64$, $L_{ls}=1024$ on SP-2 and T3D. Each data is assumed as a single precision complex number (8 bytes).	121
3.10	The input data cube and sample arrival sequence	124
3.11	Weight Computation steps for ESPrD	125
3.12	Weight Computation steps for ESPsD	126
3.13	Computational Complexities for ESPrD and ESPsD	127
3.14	Computational Requirements of ESPrD when $L = 512$	127
3.15	Computational Requirements of ESPsD when $L = 512$	128
3.16	Computational Requirements of ESPrD when $L = 4096$	128
3.17	Computational Requirements of ESPsD when $L = 4096$	129
3.18	Simple Communication Scheduling	131
3.19	Pipelined Communication Scheduling	131
3.20	Timing diagram and overlapped communication with computation	132
3.21	Estimated communication time on SP2 (o: Pipelined schedule, x: Simple Schedule) for $N,M,L = (64,64,2048), (32,32,1024), (16,16,512)$	133
3.22	Estimated communication time on T3D (o: Pipelined schedule, x: Simple Schedule) for $N,M,L = (64,64,2048), (32,32,1024), (16,16,512)$	134

Part 1

Advances in Detection and Estimation for STAP Applications with Joint Gaussian Statistics

Principal Investigator: Professor Irving S. Reed

Collaborators: Jay Scott Goldstein and Yow-Ling Gau

This part includes results on optimal techniques with joint Gaussian statistics for reduced-rank STAP applications, CFAR detection, and maximum likelihood estimation. First, in Section 1.1, the complete theoretical development of the optimal technique for reduced-rank STAP, termed the cross-spectral method, has been accomplished. The performance measure for optimality is maximizing the output signal-to-interference plus noise ratio, which is equivalent to minimizing the mean-square error of the processor. A proof of optimality has been obtained and, while counter-intuitive, the resulting solution is shown to be different from that yielded by the principal component approximation of the array covariance matrix. The rank of the resulting weight vector is reduced by the cross-spectral method, in contrast to the PCI/Eigencanceler methods. Finally, a low complexity filtering-based implementation of the cross-spectral method has been investigated, which provides excellent performance while alleviating the requirements for covariance matrix estimation and eigen-decomposition.

In Section 1.2, we design a simultaneous CFAR detection and ML estimation STAP algorithm for airborne radar. Our previous results on element-space post-Doppler STAP have been generalized and applied to the fully adaptive and beam-space post-Doppler processors. This study demonstrates that the finer resolution leads to a reduction in the detection capability. This implies that one may trade-off the accuracy of ML estimation with the performance of the CFAR detection criterion.

Finally, in Section 1.3, we develop a fast CFAR detection algorithm for STAP applications. This approach is based on the eigen-decomposition of the space-time radar covariance matrix. The method utilized by this algorithm does not require a matrix inversion, thereby reducing the computational complexity requirements of the processor.

1.1 The Cross-Spectral Metric and Optimal Rank Reduction

The problem of optimal detection is closely related to constrained Wiener filtering [1, 2, 3, 4, 5, 6]. In fact, the optimal detection criterion relies upon steering a beam in the direction and Doppler of interest, while nulling out the effects of interference and clutter in that beam. In this sense, the canonical form of the optimal detector takes the form of a partitioned processor termed the Generalized Sidelobe Canceller (GSC). The GSC was introduced by Applebaum and Chapman [7] for narrowband signals and extended to the space-time problem by Griffiths [8]. In this section, the theoretical development of optimal rank reduction for space-time adaptive processing is achieved. The cross-spectral metric [9, 10, 11, 12] is derived as the optimal technique to reduce the rank of the weight vector while maximizing the output signal-to-interference plus noise ratio (SINR). The methodology for achieving this optimization is to minimize the mean-square error of the GSC as a function of the weight vector rank. Minimizing the mean-square error and maximizing the output SINR for the GSC processor have been shown to be equivalent [6]. Indeed, it is easily shown that the weight vector which maximizes the SINR for a constrained processor and the weight vector that minimizes the mean-square error are identical.

It is important to note that other reduced-rank STAP techniques which are based on the principal component technique, such as the principal component inverse (PCI) method [13, 14] (also recently termed an Eigencanceller [15]) only form a low rank estimate of the array covariance matrix. That is to say, while the covariance matrix is approximated by only the principal eigenvectors and their corresponding eigenvalues, the resulting weight vector is still full-rank. The approach described herein differs from all other approaches in three significant ways: (1): the principal component technique provides the best low-rank approximation to the full rank covariance matrix [16]. However, this is not the correct performance measure. The correct performance measure is to find the best low-rank approximation of the covariance matrix (i.e.: the best reduced-rank subspace) for maximizing the output SINR. While counter-intuitive, these two subspaces are not the same. (2) The actual number of adaptive weights required by the PCI technique is identical to that required in the full-rank case. The cross-spectral technique results in a weight vector whose size corresponds to the reduced dimension. This means that using the cross-spectral approach to reduce the rank of the problem from some large number N to some small number M yields a weight vector of dimension M . Since the computational complexity and required hardware increases with the rank of the weight vector, this issue is paramount. (3) Finally, the development of the cross-spectral metric naturally lends itself to approximations using any unitary transform to form a basis for the space spanned by the array covariance matrix. Thus, the eigenvectors need not be used and the eigen-decomposition of the array covariance matrix need not be performed. Further, since it is possible to realize a unitary operator via fast filtering algorithms, such as the DFT and DCT, one may evaluate the cross-spectral metric and perform rank reduction without ever forming the array covariance matrix estimate from the data. This in turn implies that the rank reduction may be performed via a criterion which is based only on simple filtering operations.

1.1.1 Space-Time Adaptive Processing

Radar returns are collected in coherent processing intervals (CPI) which may be represented as a 3-D data cube as shown in Figure 1.1. The data is then processed at one range of interest, which corresponds to a slice of the CPI data cube. This slice is a $J \times K$ space-time snapshot, denoted

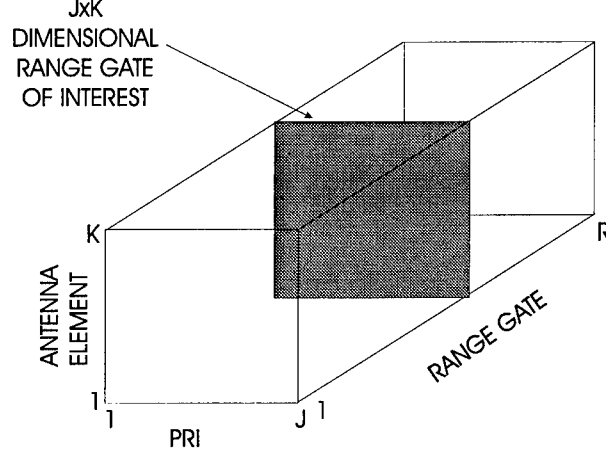


Figure 1.1: 3-Dimensional CPI Data Cube

X. The space-time snapshot can be expressed as a matrix

$$\mathbf{X} = \begin{bmatrix} x_{0,0} & x_{0,1} & \cdots & x_{0,K-1} \\ x_{1,0} & x_{1,1} & \cdots & x_{1,K-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{J-1,0} & x_{J-1,1} & \cdots & x_{J-1,K-1} \end{bmatrix} \quad (1.1)$$

where the individual elements $x_{j,k}$ correspond to the data from the j -th pulse repetition interval (PRI) and the k -th sensor element [17, 18].

The 2-D space-time data structure consists of element space information and PRI space Doppler information. Each return \mathbf{X} is composed of components due to the target, the interference sources or jammers, clutter, and white noise:

$$\mathbf{X} = \mathbf{X}_t + \mathbf{X}_i + \mathbf{X}_c + \mathbf{X}_w. \quad (1.2)$$

The noise in the space-time snapshot is the thermal noise present at the sensor elements, which is spatially and temporally white. Each of the other components of the radar return are now addressed.

The target is assumed to be a point target, present in only one range bin. The $J \times K$ space-time snapshot corresponding to the target is then given by

$$\mathbf{X}_t = \alpha_t \mathbf{b}(\omega_t) \mathbf{a}(\vartheta_t)^H, \quad (1.3)$$

where $\mathbf{a}(\vartheta_t)$ is the $K \times 1$ spatial steering vector, $\mathbf{b}(\omega_t)$ is the $J \times 1$ temporal steering vector, and α_t is the amplitude of the target.

The interference environment consists of several sources which are assumed to be temporally white (barrage jammers spread over all Doppler frequencies at a particular azimuth). The $J \times K$ space-time snapshot corresponding to the jammers is then given by

$$\mathbf{X}_i = \alpha_i \mathbf{A}(\vartheta_i)^H, \quad (1.4)$$

where α_i is the $J \times I$ matrix of jammer amplitude for the J PRIs of each of the I interference sources and $\mathbf{A}(\vartheta_i)$ is the $K \times I$ matrix whose row are the spatial steering vectors for each interference source.

The clutter returns when considering a target at a given range will come from all areas within a common range about the radar platform. Consider returns from a circular ring about the platform. If each spot on the clutter ring is considered to be a point target, its space-time snapshot will take the form

$$\mathbf{X}_{c_p} = \alpha_{c_p} \mathbf{b}(\omega_{c_p}) \mathbf{a}(\vartheta_{c_p})^H, \quad (1.5)$$

where ω_{c_p} and ϑ_{c_p} are the spatial and Doppler frequencies of the clutter return from the p -th patch, and α_{c_p} is the power from the p -th clutter patch. The clutter is assumed to be stationary and it has a velocity relative to the platform housing the radar which is determined by the azimuth angle between the array and the clutter patch. For simulation purposes, the clutter returns are assumed to come from the entire clutter ring, which is divided into 360 evenly spaced sectors, and summed so that

$$\mathbf{X}_c = \sum_{\phi_{c_p}=1}^{360} \alpha_{c_p} \mathbf{b}(\beta \vartheta_{c_p}) \mathbf{a}(\vartheta_{c_p})^H, \quad (1.6)$$

where β is the constant of linearity between the Doppler frequency and the spatial frequency and

$$\vartheta_{c_p} = \frac{d}{\lambda_o} \cos \theta_c \sin \phi_{c_p}, \quad (1.7)$$

with d being the element spacing, λ_o being the wavelength, and θ_c being the fixed elevation angle.

Each of the components of the space-time snapshots are assumed to be uncorrelated. These component matrices are stacked column-wise to form the $KJ \times 1$ vector $\mathbf{x} = \mathbf{x}_t + \mathbf{x}_i + \mathbf{x}_c + \mathbf{x}_w$. The resulting space-time covariance matrix is the $KJ \times KJ$ matrix \mathbf{R}_x :

$$\mathbf{R}_x = \mathbf{E}[\mathbf{x}\mathbf{x}^H] = \mathbf{R}_t + \mathbf{R}_i + \mathbf{R}_c + \mathbf{R}_w, \quad (1.8)$$

where \mathbf{R}_t , \mathbf{R}_i , \mathbf{R}_c , and \mathbf{R}_w are the target signal, interference, clutter, and white noise covariance matrices, respectively.

1.1.2 The GSC Form Processor

The structure of the processor under consideration is the GSC partitioned form processor. The received snapshot is of dimension KJ , which is the product of the number of sensor elements and the number of Doppler frequency bins. The GSC is implemented by partitioning the received data with filters \mathbf{w}_c and \mathbf{W}_s . The conventional beamforming filter \mathbf{w}_c is a vector which enforces the look-direction constraint in angle and Doppler. This desired signal is blocked from the adaptive processor through the signal blocking matrix \mathbf{W}_s , which in general is of dimension $N \times KJ$ where $N < KJ$. For example, in the single linear constraint case, $N = KJ - 1$. The full row rank matrix \mathbf{W}_s is composed of rows \mathbf{a}_i such that $\mathbf{a}_i \mathbf{w}_c = 0$ for $i = 1, 2, \dots, KJ$.

As previously described, the $KJ \times 1$ -dimensional received signal vector present on the antenna elements at time k is denoted by $\mathbf{x}(k)$ and the associated $KJ \times KJ$ received input data covariance matrix is denoted by $\mathbf{R}_x = \mathbf{E}[\mathbf{x}(k)\mathbf{x}^H(k)]$. The N -dimensional noise subspace data vector $\mathbf{x}_s(k)$,

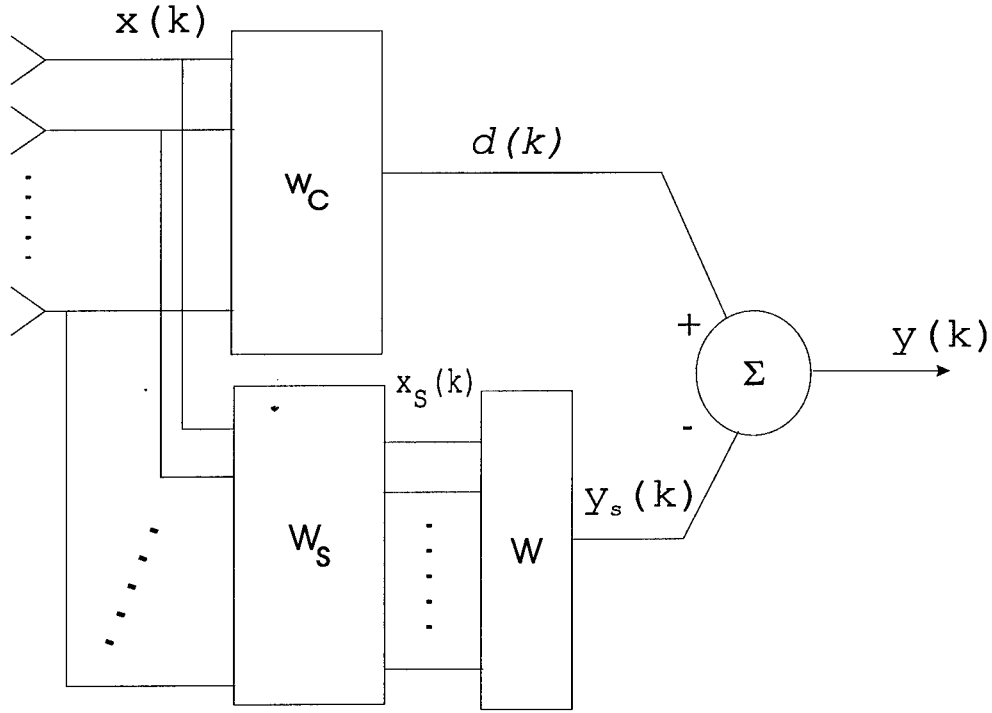


Figure 1.2: The full-rank GSC MVDR processor.

the scalar beamformed output $d(k)$, and the scalar beamformed noise estimator $y_s(k)$ are given by

$$\begin{aligned} \mathbf{x}_s(k) &= \mathbf{W}_s \mathbf{x}(k), \\ d(k) &= \mathbf{w}_c^H \mathbf{x}(k), \\ y_s(k) &= \mathbf{w}^H \mathbf{x}_s(k), \end{aligned} \quad (1.9)$$

where \mathbf{w} is the N -dimensional weight vector. Finally, the array output is

$$y(k) = (\mathbf{w}_c^H - \mathbf{w}^H \mathbf{W}_s) \mathbf{x}(k). \quad (1.10)$$

Evidently, the mean-square error of the processor is given by the mean-square value of $y(k)$.

The GSC array in Figure 1.2 converges to the discrete-time Wiener-Hopf solution given by

$$\mathbf{w} = \mathbf{R}_{x_s}^{-1} \mathbf{r}_{x_s d}, \quad (1.11)$$

where the observation data covariance matrix is expressed as

$$\mathbf{R}_{x_s} = \mathbf{E}[\mathbf{x}_s(k) \mathbf{x}_s^H(k)] = \mathbf{W}_s \mathbf{R}_x \mathbf{W}_s^H, \quad (1.12)$$

and the cross-correlation vector between the noise subspace data vector and the beamformer output is given by

$$\mathbf{r}_{x_s d} = \mathbf{E}[\mathbf{x}_s(k) d^*(k)] = \mathbf{W}_s \mathbf{R}_x \mathbf{w}_c. \quad (1.13)$$

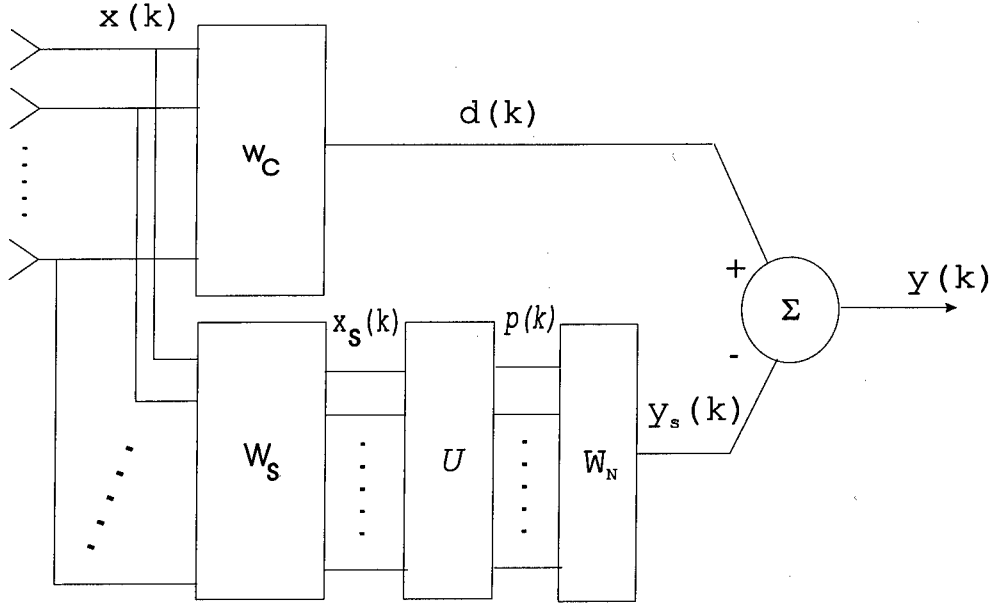


Figure 1.3: The full-rank GSC MVDR processor in principal coordinates.

The minimum mean-square error, denoted P , is found by substituting (1.11) into (1.10) and evaluating the mean-square value of $y(k)$:

$$P = \mathbb{E}[|y(k)|^2] = \sigma_d^2 - \mathbf{r}_{x_s d}^H \mathbf{R}_{x_s}^{-1} \mathbf{r}_{x_s d}, \quad (1.14)$$

where $\sigma_d^2 = \mathbf{w}_c^H \mathbf{R}_{xx} \mathbf{w}_c$ is the variance of the conventional beamformer output.

The observation data covariance matrix \mathbf{R}_{x_s} may be expressed in terms of its eigenvectors and eigenvalues as

$$\mathbf{R}_{x_s} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^H, \quad (1.15)$$

where \mathbf{U} is a unitary $N \times N$ matrix composed of the eigenvectors $\{\mathbf{v}_i\}_{i=1}^N$ and $\mathbf{\Lambda}$ is the diagonal matrix of associated eigenvalues $\{\lambda_i\}_{i=1}^N$. In terms of the principal coordinates of the problem, we define the process $\mathbf{p}(k) = \mathbf{U}^H \mathbf{x}_s(k)$. A normal component covariance matrix \mathbf{R}_p , cross-correlation vector \mathbf{r}_{pd} , and Wiener filter \mathbf{w}_N are defined now as follows:

$$\begin{aligned} \mathbf{R}_p &= \mathbb{E}[\mathbf{p}(k) \mathbf{p}^H(k)] = \mathbf{U}^H \mathbf{R}_{x_s} \mathbf{U} = \mathbf{\Lambda}, \\ \mathbf{r}_{pd} &= \mathbb{E}[\mathbf{p}(k) d^*(k)] = \mathbf{U}^H \mathbf{r}_{x_s d}, \\ \mathbf{w}_N &= \mathbf{R}_p^{-1} \mathbf{r}_{pd} = \mathbf{U}^H \mathbf{w}. \end{aligned} \quad (1.16)$$

The GSC in these normal coordinates, depicted in Figure 1.3, is equivalent to the GSC in Figure 1.2 in terms of its steady-state characteristics. The array output of the GSC in normal coordinates is given by

$$y = (\mathbf{w}_c^H - \mathbf{w}_N^H \mathbf{U}^H \mathbf{W}_s) \mathbf{x}(k) = (\mathbf{w}_c^H - \mathbf{w}^H \mathbf{W}_s) \mathbf{x}(k). \quad (1.17)$$

Note that the MMSE,

$$P = \sigma_d^2 - \mathbf{r}_{pd}^H \mathbf{R}_p^{-1} \mathbf{r}_{pd} = \sigma_d^2 - \mathbf{r}_{x_s d}^H \mathbf{R}_{x_s}^{-1} \mathbf{r}_{x_s d}, \quad (1.18)$$

is conserved by any unitary transformation, including that realized by the operator \mathbf{U} .

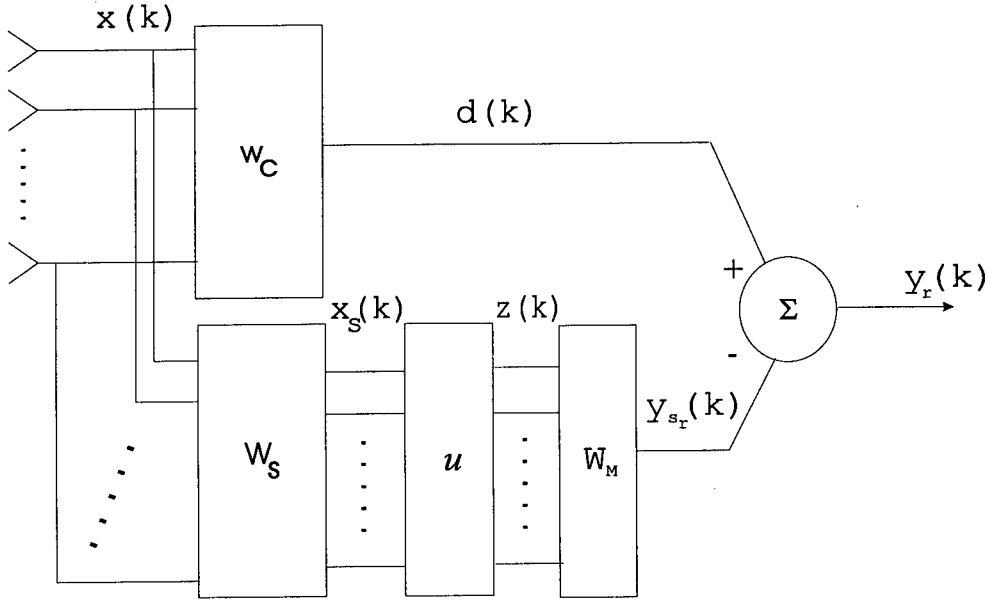


Figure 1.4: The reduced-rank GSC MVDR processor.

1.1.3 Partially Adaptive Processing

The problem of reducing the degrees of freedom for an array processor involves selecting a subset or some combination of the elements to be adaptively weighted. For notational purposes, we let the space spanned by the columns of the fully adaptive array covariance matrix be denoted by \mathcal{C}^N , implying that the observation covariance matrices \mathbf{R}_{x_s} and \mathbf{R}_p are of dimension $N \times N$ and the vectors $\mathbf{r}_{x_{sd}}$, \mathbf{r}_{pd} , \mathbf{w} , and \mathbf{w}_N are $N \times 1$ -dimensional vectors. The partially adaptive GSC shown in Figure 1.4 utilizes an $N \times M$ transformation operator \mathbf{U} , in place of \mathbf{U} in Figure 1.3, to form the M -dimensional reduced-rank observation data vector

$$\mathbf{z}(k) = \mathbf{U}^H \mathbf{x}_s(k), \quad (1.19)$$

where $M < N$. The associated $M \times M$ reduced-rank covariance matrix is given by

$$\mathbf{R}_z = \mathbf{U}^H \mathbf{R}_{x_s} \mathbf{U}. \quad (1.20)$$

The data vector $\mathbf{z}(k)$ is then processed by the reduced-rank weight vector \mathbf{w}_M , which is of dimension $M \times 1$. It is the selection of the rank reducing operator \mathbf{U} which serves as the present topic of interest.

The most popular technique for subspace selection is based upon the principal components method [19, 20, 21]. This method determines the singular value decomposition of the $N \times N$ -dimensional covariance matrix \mathbf{R}_{x_s} and selects the M largest eigenvectors (those corresponding to the largest eigenvalues) to form the M -dimensional eigen-subspace $\Psi \subset \mathcal{C}^N$ in which the adaptive processor operates. However, this technique does not directly consider the MMSE performance measure, which is a function of not only the space spanned by the noise covariance matrix but also of the cross-correlation between the desired signal and the noise process. It is noted that Byerly [22] discovered that the eigenvectors corresponding to the largest eigenvalues were not

necessarily the best selection, but there was no derivation provided and the approach obtained herein provides a more general solution.

Following Scharf [3, ch. 8], we examine the problem of reducing the rank of the Wiener filter. In Figure 1.4, the rank reducing $N \times M$ transformation matrix \mathbf{U} is composed of some M columns from \mathbf{U} which are to be selected. The operator \mathbf{U} is constrained therefore to be a subset of M of the N possible eigenvectors of \mathbf{R}_{x_s} . This particular constraint allows a direct comparison with the principal component technique, which chooses the rank reducing transform to be composed of those M eigenvectors corresponding to the largest M eigenvalues. Thus, the particular problem at hand is to choose the subspace spanned by a set of M eigenvectors out of the N available such that the resulting M -dimensional Wiener filter yields the lowest MMSE out of all $\binom{N}{M}$ possible combinations of eigenvectors.

Now denote the reduced-rank processor output by $y_r(k)$ and the reduced-rank noise estimator by $y_{sr}(k)$. This is illustrated in Figure 1.4. A study of Figures 1.3 and 1.4 suggest that the reduced-rank processor output may be expressed as

$$y_r(k) = \begin{bmatrix} 1 & \mathbf{w}_N^H \mathbf{U}^H - \mathbf{w}_M^H \mathbf{U}^H \end{bmatrix} \begin{bmatrix} y(k) \\ \mathbf{x}_s(k) \end{bmatrix}. \quad (1.21)$$

Denote the weight error vector between the full-rank weight vector and its reduced-rank version by

$$\mathbf{e} = \mathbf{U} \mathbf{w}_N - \mathbf{U} \mathbf{w}_M. \quad (1.22)$$

The mean-square value of the reduced-rank processor output $y_r(k)$ in this notation is computed to be

$$\mathbf{E}[|y_r(k)|^2] = P + \mathbf{e}^H \mathbf{R}_{x_s} \mathbf{e}, \quad (1.23)$$

where P is the full-rank MMSE and $P + \mathbf{e}^H \mathbf{R}_{x_s} \mathbf{e}$ is the reduced-rank MMSE. It is now desired to choose the rank reducing operator \mathbf{U} in such a manner that \mathbf{w}_M minimizes the additional mean-square error incurred by rank reduction, namely the scalar term $\mathbf{e}^H \mathbf{R}_{x_s} \mathbf{e}$ in (1.23), as follows:

$$\min_{\mathbf{w}_M} [\mathbf{e}^H \mathbf{R}_{x_s} \mathbf{e}] = \min_{\mathbf{w}_M} \left[\left(\mathbf{w}_N^H \mathbf{R}_p^{1/2} - \mathbf{w}_M^H \mathbf{U}^H \mathbf{R}_{x_s}^{1/2} \right) \left(\mathbf{w}_N^H \mathbf{R}_p^{1/2} - \mathbf{w}_M^H \mathbf{U}^H \mathbf{R}_{x_s}^{1/2} \right)^H \right]. \quad (1.24)$$

Then evidently the best solution for (1.24) is to choose \mathbf{U} such that $\mathbf{w}_M^H \mathbf{U}^H \mathbf{R}_{x_s}^{1/2} = \mathbf{w}_M^H \mathbf{R}_z^{1/2}$ is the best low rank approximation to the vector $\mathbf{w}_N^H \mathbf{R}_p^{1/2}$.

The Wiener-Hopf relationship for the full-rank case,

$$\mathbf{R}_p \mathbf{w}_N = \mathbf{r}_{pd}, \quad (1.25)$$

implies that

$$\begin{aligned} \mathbf{w}_N^H \mathbf{R}_p^{1/2} &= \mathbf{r}_{pd}^H \mathbf{R}_p^{-1/2} = \mathbf{r}_{x_s d}^H \mathbf{U} \mathbf{A}^{-1/2} \\ &= \begin{bmatrix} \frac{\mathbf{v}_1^H \mathbf{r}_{x_s d}}{\sqrt{\lambda_1}} \\ \frac{\mathbf{v}_2^H \mathbf{r}_{x_s d}}{\sqrt{\lambda_2}} \\ \vdots \\ \frac{\mathbf{v}_N^H \mathbf{r}_{x_s d}}{\sqrt{\lambda_N}} \end{bmatrix}. \end{aligned} \quad (1.26)$$

Thus, in order to make the vector $\mathbf{w}_M^H \mathbf{U}^H \mathbf{R}_{x_s}^{1/2}$ be the best low rank approximation to the vector $\mathbf{w}_N^H \mathbf{R}_p^{1/2}$, it is necessary to rank order the terms in (1.26) by their magnitude. Then the rank reducing operator \mathbf{U} is selected by choosing those M eigenvectors which correspond with the largest M values of the sequence of non-negative terms

$$\left| \frac{\mathbf{v}_i^H \mathbf{r}_{x_{sd}}}{\sqrt{\lambda_i}} \right|^2, \quad (1.27)$$

for $i = 1, 2, \dots, N$. With this selection, the columns of the reduced-rank covariance matrix \mathbf{R}_z span the M -dimensional cross-spectral subspace $\Omega \subset \mathcal{C}^N$ to provide the lowest MMSE of any M -dimensional subspace which is spanned by M of the N columns of \mathbf{U} . It is noted that this solution is similar to the SVD technique described in [3, ch. 8.4]. Also it is of interest physically that the term in (1.27) measures the cross-spectral energy projected along the i -th eigenvector.

Hence the reduced-rank Wiener filter in the subspace Ω is given by

$$\mathbf{w}_M = (\mathbf{U}^H \mathbf{R}_{x_s} \mathbf{U})^{-1} \mathbf{U}^H \mathbf{r}_{x_{sd}} = \mathbf{\Lambda}_M^{-1} \mathbf{U}^H \mathbf{r}_{x_{sd}}, \quad (1.28)$$

where $\mathbf{\Lambda}_M$ is the diagonal matrix composed of the M eigenvalues corresponding to the eigenvectors which form \mathbf{U} . Clearly, the subspace Ω spanned by the columns of eigenvectors corresponding with the M largest values of the cross-spectral metric is not the same as the subspace Ψ spanned by the eigenvectors corresponding with the M largest eigenvalues. This means that the Wiener filter in the cross-spectral subspace Ω yields a lower MMSE than the Wiener filter in the subspace Ψ .

To demonstrate that the cross-spectral metric is optimal for each rank $M \leq N$, consider the decomposition of the MMSE performed by the full-rank matrix of eigenvectors \mathbf{U} . The full-rank MMSE is given by

$$\begin{aligned} P &= \sigma_d^2 - \mathbf{r}_{pd}^H \mathbf{R}_p^{-1} \mathbf{r}_{pd} = \sigma_d^2 - \mathbf{r}_{x_{sd}}^H \mathbf{U} \mathbf{\Lambda}^{-1} \mathbf{U}^H \mathbf{r}_{x_{sd}} \\ &= \sigma_d^2 - \sum_{i=1}^N \frac{|\mathbf{v}_i^H \mathbf{r}_{x_{sd}}|^2}{\lambda_i}. \end{aligned} \quad (1.29)$$

Finally, a comparison of Equations (1.27) and (1.29) demonstrate that the selection of the subspace which provides the largest cross-spectral contribution also results in the lowest MMSE as a function of the rank of the Wiener filter. Since minimizing the mean-square error and maximizing the output SINR are equivalent for the constrained array, the reduced-rank Wiener filter found via the cross-spectral metric maximizes the output SINR as a function of the rank of the filter.

1.1.4 Example Using the Mountaintop Radar Parameters

A simulation is now considered which utilizes the parameters of the Mountaintop radar to compare the performance of the reduced-rank processors to that of the full-rank, joint domain optimal processor. In particular, the output SINR for the full-rank processor, the eigen-subspace reduced-rank processor and two cross-spectral subspace reduced-rank processors will be computed. Finally, the output SINR as a function of Wiener filter rank will be presented for these processors and contrasted to that resulting from the standard technique of simply choosing the largest principal components of the noise covariance matrix.

The second cross-spectral subspace processor, mentioned in the preceeding paragraph, utilizes the DCT in place of the eigenvectors of the array covariance matrix. This new frequency domain processor has an extremely low computational complexity. The trade-off is a minor loss in performance, to be demonstrated shortly.

A. Description of the Mountaintop Radar and the Signal Environment

The Mountaintop radar employs the Radar Surveillance Technology Experimental Radar (RSTER) and the Inverse Displaced Phase Center Array (IDPCA) co-located at the same site [23]. It is assumed that the radar is in the RSTER-90 configuration and receive-only mode. The transmit frequency is 450 MHz. This radar consists of $K=14$ elements and $J=16$ pulses in the PRI. The elevation angle θ is fixed (pre-beamformed), and the azimuth angle ϕ is the only free parameter.

The processing is performed over one range gate of interest, where it is assumed that this range gate contains the target signal. Thus, the number of inputs to the adaptive processor is $KJ = 224$. The GSC processor beamforms the target signal in one branch and blocks the target signal from the other branch, as described earlier in this section. The noise subspace data vector is then of dimension $N = KJ - 1 = 223$. The corresponding full-rank GSC weight vector is of dimension 223×1 .

For the purpose of this analysis, three barrage jammers and clutter compose the interference environment, and one target signal is present. The radar platform is assumed to be at a height of 500 meters, and the platform velocity is 500 meters per second. The target, which is also at a height of 500 meters, has a range of 22 kilometers and an azimuth of -30° . The target velocity is 250 meters per second. This target has a signal-to-noise ratio (SNR) of 0 dB, represents a very small constant radar cross-section target. The three jammers each have an SNR of 40 dB and azimuth angles of -60° , 30° and 60° , respectively. The clutter-to-noise ratio is 45.6 dB.

B. Analysis and Results

We now examine the performance of the fully adaptive GSC, the partially adaptive eigen-subspace GSC, and the partially adaptive cross-spectral subspace GSC processors. To evaluate the performance of the subspace selection techniques, the dimension of the adaptive processor will be reduced from $N = 223$ weights to $M = 50$. The dimension of the noise subspace eigenstructure is 77, which is supposedly the lower bound for rank reduction [19, 20].

The power spectrum of the data, averaged over 500 snapshots, is presented in Figure 1.5, from which it is evident that the target signal is not discernible. Figure 1.6 presents the full rank Wiener filter power spectrum, from which it is seen that the optimal space-time weight vector of dimension 223 is capable of attenuating all jammers and clutter while passing the target signal. The full-rank Wiener filter SINR is 23.28 dB. Figure 1.7 depicts the rank 50 eigen-subspace Wiener filter. The clutter is attenuated, but most of the power from the three jammers is passed with relatively high gain, particularly the jammer at an azimuth angle of 30° . The output SINR for the eigen-subspace Wiener filter is 7.65 dB, reflecting a loss of 15.63 dB. The power spectrum of the rank 50 cross-spectral subspace Wiener filter is depicted in Figure 1.8. It can be seen that by simply selecting a different set of 50 eigenvectors, the resulting power spectrum closely approximates the full rank optimal and all clutter and jammers are attenuated. The resulting output SINR is 23.20 dB, reflecting a loss of only 0.08 dB in reducing the rank from 223 to 50.

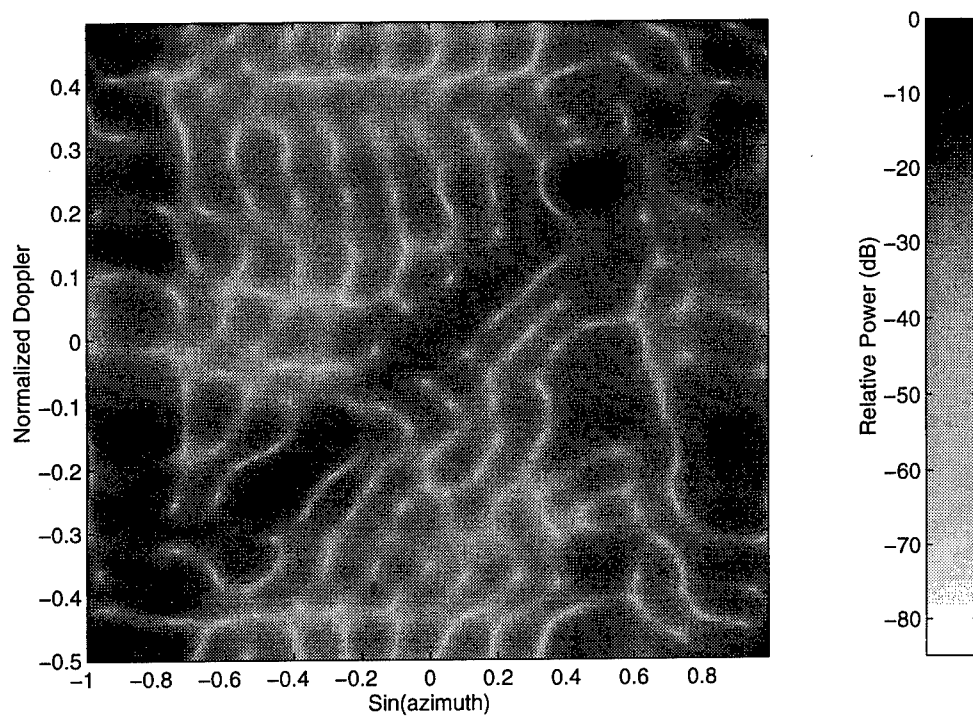


Figure 1.5: The power spectrum of the data

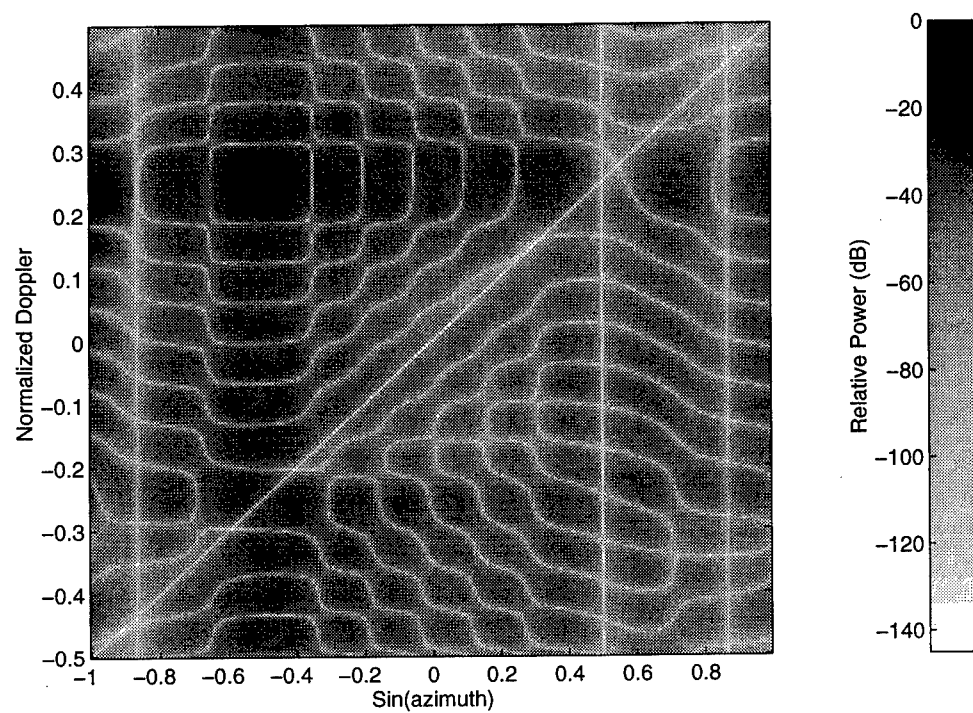


Figure 1.6: The full-rank Wiener filter power spectrum

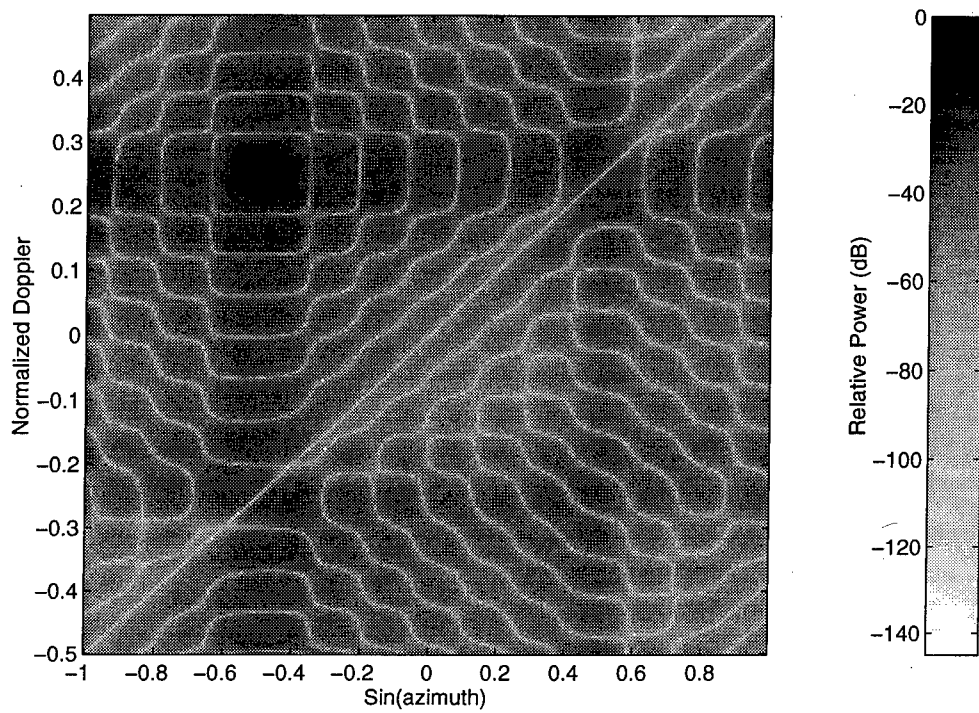


Figure 1.7: The reduced-rank eigen-subspace Wiener filter power spectrum

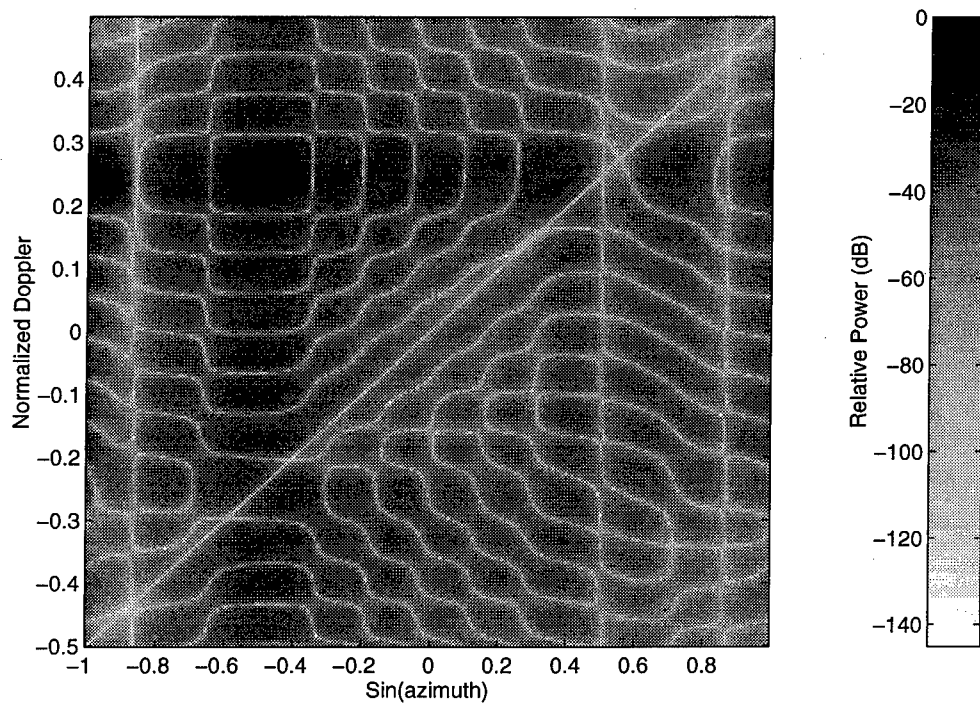


Figure 1.8: The reduced rank cross-spectral subspace Wiener filter power spectrum

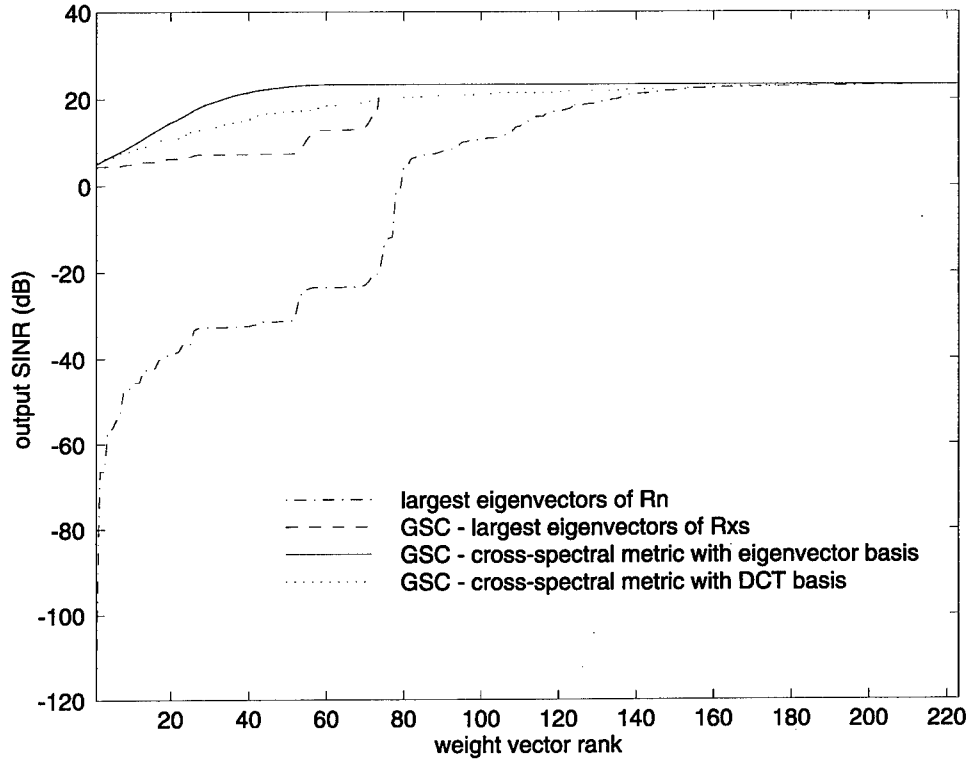


Figure 1.9: The output SINR of the reduced-rank processors as a function of the rank of the data

The performance of these two reduced-rank processors may be evaluated by plotting the output SINR as a function of Wiener filter rank. In addition, we are interested in comparing two other reduced-rank processors.

The first additional reduced-rank processor under consideration is provided via the PCI technique. This method estimates the full rank noise covariance matrix $\mathbf{R}_n = \mathbf{R}_i + \mathbf{R}_c + \mathbf{R}_w$ by the largest principal components to form $\hat{\mathbf{R}}_n$. For the purpose of this analysis, the corresponding eigenvalues are retained in the estimate as opposed to making the common approximation that they have a value of unity. The resulting equivalent direct form array has a constrained Wiener filter $\hat{\mathbf{w}}$ given by

$$\hat{\mathbf{w}} = \frac{\hat{\mathbf{R}}_n^{-1} \mathbf{w}_c}{\mathbf{w}_c^H \hat{\mathbf{R}}_n^{-1} \mathbf{w}_c}. \quad (1.30)$$

and the array output is $y_r(k) = \hat{\mathbf{w}}^H \mathbf{x}(k)$. Note that the weight vector is full-rank.

The second additional reduced-rank processor under consideration uses the GSC form array with the DCT unitary operator for subspace selection. The use of the DCT matrix \mathbf{Q} in place of the matrix of eigenvectors \mathbf{U} in Figure 1.3 permits the use of an approximation to the cross-spectral metric. Thus, the rank reducing transformation \mathbf{U} in Figure 1.4 is composed of some M columns of the DCT matrix \mathbf{Q} to be selected by a modified version of the cross-spectral metric. This modified metric replaces the numerator of (1.27) with the mean-square value of the product formed by the i -th DCT output and the beamformed signal $d(k)$. The denominator is replaced with the power of the data in the i -th DCT frequency bin.

Figure 1.9 depicts the output SINR of these four reduced-rank processors as a function of the rank of the data. For the three GSC processors, the rank of the Wiener filter and the rank of the data are identical. Thus, this plot depicts the SINR as a function of the Wiener filter rank in these cases, whereas the curve corresponding to the PCI method (utilizing the low-rank covariance estimate) requires a rank 224 weight vector for each value of the rank along the x-axis. It may be seen from Figure 1.9 that the cross-spectral metric with the eigenvector basis performs the best, obtaining the optimal SINR at a Wiener filter rank of approximately 56. The largest eigenvector technique using the GSC requires a processor of approximately the dimension of the noise subspace eigenstructure, corresponding to 77 weights, to obtain the optimal solution. The cross-spectral metric approximation with the DCT basis performs better than the GSC eigen-subspace processor until the rank of the weight vector reaches the dimension of the noise subspace eigenstructure. The PCI-type processor, which forms a low-rank estimate of the noise covariance matrix, performs the worst. The PCI approach does not obtain the optimal solution until the covariance estimate nearly utilizes all of the principal components. In addition, the PCI processor performs very poorly until the number of principal components included in the covariance estimate reaches the rank of the noise subspace eigenstructure. Only at that point does the PCI technique begin an exponential convergence to the full-rank SINR; conversely, both cross-spectral approaches converge exponentially along their entire trajectory.

1.1.5 Conclusions

A cross-spectral metric for subspace selection in partially adaptive array processing is derived and a proof provided to show that this metric is the optimal performance measure to use in deciding which eigenvectors to keep for rank reduction. Also it is demonstrated that a GSC array, operating in the subspace selected by the cross-spectral subspace estimator, exceeds the performance realized by operation in a subspace which is based upon the principal components method. The cross-spectral subspace provides a better performance than the eigen-subspace when the conventional beamformer pattern provides any attenuation of the interference. An example and analysis are provided on the assumption of exact signal knowledge, although estimated statistics also may be utilized in conjunction with an adaptive algorithm. Finally, the performance of these reduced-rank techniques are compared to the PCI method and a low computational complexity, DCT based, approximation to the cross-spectral metric. It is demonstrated that the GSC form array methods outperform the PCI technique and result in reduced-rank weight vectors, whereas the PCI method always requires a full-rank filter.

1.2 A Simultaneous CFAR Detection and ML Estimation STAP Algorithm for Airborne Radar

The algorithm in this section provides both a CFAR detection and a maximum likelihood (ML) Doppler-bearing estimator of a target in a background of unknown Gaussian noise. A target is detected, and its parameters estimated within each range gate by evaluating a statistical test for each Doppler-angle cell and by selecting the cell with maximum output and finally comparing it with a threshold. The CFAR performance of the proposed processor is analyzed by the use of the sample matrix inversion (SMI) method and is evaluated in the cases of a fully adaptive STAP and two partially adaptive STAP's. The performances of these criteria show that the probability

of detection is a function only of the sample size K used to estimate the covariance matrix and a generalized signal-to-noise ratio. The choice of the number K is a trade-off between performance and computational complexity. The performance curves demonstrate that the finer the resolution is, the poorer the detection capability. This means that one can trade off the accuracy of ML estimation with the performance of the CFAR detection criterion.

1.2.1 Introduction

A requirement of future airborne radar is to detect targets in an interference background which is comprised of both clutter and jamming. Conventional radars are capable of detecting targets in the time domain (range) and the frequency domain (Doppler frequency). Space-time adaptive processing (STAP) radars provide an additional domain (space) for the detection of targets. STAP is a two-dimensional adaptive filtering algorithm which uses both temporal and spatial filtering to suppress interference.

In the receive mode, radar target echoes are received usually within a transmit beamwidth. But, depending on the array aperture and, more importantly, on the target range, the angular sector encompassed by the mainlobe of the receive beam may correspond to a fairly large swath of space. Thus, many measurements and an interpolation are needed to more precisely determine the bearing of the target.

Traditionally, the process of target detection and estimation is carried out in sequence by different operational techniques. However, in this section STAP target detection and ML parameter estimation are obtained simultaneously, except for a final interpolation, with a single criterion. First, the fully adaptive test (defined in [18] as fully adaptive if the process refers to both the spatial and temporal domain) is based on the assumption of known signal parameters, range, Doppler, the elevation and azimuth angles and a known noise covariance matrix. Then, this test is normalized so that it has a unit variance. Finally, the test is extended to find the maximum response of the Doppler-angle filter bank. The system block diagram of this fully adaptive STAP is shown in Figure 1.10. To simplify the analysis, the responses of the set of Doppler-angle filters are assumed to be mutually orthogonal. Under such an assumption the false-alarm probability and the probability of detection with a known covariance matrix are found analytically in this section.

Also, the statistical performance of this detector is extended and analyzed by replacing the noise covariance matrix by a maximum likelihood (ML) estimate of the noise covariance which is based on actual data. It is demonstrated here that this makes the above test into a powerful CFAR detection test which simultaneously estimates both the Doppler frequency and bearing angle of the target.

However, the computational complexity for the required number of degrees of freedom makes the above fully adaptive STAP difficult to realize in practice [18]. Two classes of reduced-dimension estimator-detectors, which are called partially adaptive STAP, are also developed and analyzed in this section. A partially adaptive STAP is defined to be a processor which is not fully adaptive simultaneously in both the spatial and temporal domain. The partially adaptive element-space post-Doppler STAP is shown in Figure 1.11, and the partially adaptive beam-space pre-Doppler STAP is shown in Figure 1.12. Finally, these two suboptimal STAP criteria are compared with the above-described fully adaptive STAP.

1.2.2 Formulation of the Problem

The system under consideration is a coherent pulsed Doppler radar. For simplicity the radar antenna is assumed to be a uniformly spaced linear array antenna consisting of N elements. The radar transmits a coherent burst of M pulses. For each pulse repetition interval (PRI) L range samples are collected. With M pulses and N receiver channels, the received data for one coherent-processing interval (CPI) comprises LMN complex baseband samples, called the LMN data cube.

For the present it is assumed that a target remains in one range gate during a CPI. The snapshot of the MN samples of data for the range gate, which contains target signal, is the so-called primary data. The secondary data consists of the outputs of K range gates in the vicinity of the range gate of a target to be detected. This set of K range gates consists of a subset of the remaining $(L-1)$ range gates.

The noise components of the data vectors consist of thermal noise, clutter and jamming which are modeled as zero-mean complex Gaussian random vectors. The noise components of each range gate are assumed to share a common covariance matrix and to be mutually independent.

For the analysis it is desirable to assume that each filter is statistically independent. This is achieved approximately by the proper spacing between adjacent filters and beams.

1.2.3 Derivation of the Test Statistic (when Noise Covariance is Known)

Let \mathbf{x} be a $MN \times 1$ snapshot of a given range gate, i.e. the observation vector,

$$\mathbf{x} = [x_{11}, x_{12}, \dots, x_{1N}, x_{21}, x_{22}, \dots, x_{2N}, \dots, x_{M1}, x_{M2}, \dots, x_{MN}]^T, \quad (1.31)$$

where x_{ij} represents the sample from the i^{th} pulse and the j^{th} antenna element.

Two hypotheses are postulated :

$$H_0: \mathbf{x} = \mathbf{n}, \quad (1.32)$$

$$H_1: \mathbf{x} = \mathbf{s} + \mathbf{n}, \quad (1.33)$$

where \mathbf{n} is a $MN \times 1$ Gaussian noise vector with known covariance matrix \mathbf{R} and \mathbf{s} is a known $MN \times 1$ target vector with phase angle ϑ_n , Doppler frequency ω_m and a complex amplitude α with random phase ϕ . Thus \mathbf{s} can be represented by the MN dimensional vector,

$$\mathbf{s} = \alpha \mathbf{b}(\omega_m) \otimes \mathbf{a}(\vartheta_n) = \alpha \mathbf{v}(\omega_m, \vartheta_n), \quad (1.34)$$

where $\mathbf{b}(\omega_m)$ is a $M \times 1$ temporal steering vector, $\mathbf{a}(\vartheta_n)$ is a $N \times 1$ spatial steering vector, \otimes denotes Kronecker product and $\mathbf{v}(\omega_m, \vartheta_n) = \mathbf{b}(\omega_m) \otimes \mathbf{a}(\vartheta_n)$ is a $MN \times 1$ steering vector.

Since the starting phase ϕ is a random variable, the random vector \mathbf{x} is not strictly Gaussian. However, when conditioned on ϕ , \mathbf{x} is complex Gaussian under both hypotheses H_0 and H_1 . Thus the conditional probability densities of \mathbf{x} are

$$P(\mathbf{x}|H_1, \phi) = \frac{\exp\{-[\mathbf{x} - \alpha \mathbf{v}(\omega_m, \vartheta_n)]^H \mathbf{R}^{-1} [\mathbf{x} - \alpha \mathbf{v}(\omega_m, \vartheta_n)]\}}{\pi^N |\mathbf{R}|}, \quad (1.35)$$

$$P(\mathbf{x}|H_0) = \frac{1}{\pi^N |\mathbf{R}|} \exp\{-\mathbf{x}^H \mathbf{R}^{-1} \mathbf{x}\}, \quad (1.36)$$

where $\alpha = |\alpha|e^{j\phi}$ with the starting phase ϕ , which is uniformly distributed between $[0, 2\pi]$, and $\mathbf{R} = E\{\mathbf{nn}^H\}$.

The likelihood ratio test Λ_{mn} of the densities in (1.35) and (1.36) is

$$\begin{aligned}\Lambda_{mn} &= \frac{P(\mathbf{x}|H_1)}{P(\mathbf{x}|H_0)} = \frac{\int_0^{2\pi} P(\mathbf{x}|H_1, \phi) \frac{1}{2\pi} d\phi}{P(\mathbf{x}|H_0)} \\ &= \int_0^{2\pi} \exp \left\{ 2\text{Re} \left[\alpha^* \mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1} \mathbf{x} \right] - |\alpha|^2 \mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1} \mathbf{v}(\omega_m, \vartheta_n) \right\} \frac{1}{2\pi} d\phi \\ &= \exp \left\{ -|\alpha|^2 \mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1} \mathbf{v}(\omega_m, \vartheta_n) \right\} I_0 \left(2|\alpha| \left| \mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1} \mathbf{x} \right| \right) \underset{H_0}{\overset{H_1}{\geq}} T_0 \quad (1.37)\end{aligned}$$

where $I_0(x)$ is the modified Bessel function of the first kind. Since a covariance matrix is non-negative definite, i.e. $\mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1} \mathbf{v}(\omega_m, \vartheta_n) \geq 0$, the exponential term on the left hand side of the above equation can be absorbed into the threshold. Hence the above test becomes

$$I_0 \left(2|\alpha| \left| \mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1} \mathbf{x} \right| \right) \underset{H_0}{\overset{H_1}{\geq}} T_0 \exp \left\{ |\alpha|^2 \mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1} \mathbf{v}(\omega_m, \vartheta_n) \right\} = T_1 \quad (1.38)$$

Since $I_0(x)$ is a monotonically increasing function of positive x , this test further reduces to the following test :

$$\left| \mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1} \mathbf{x} \right|^2 \underset{H_0}{\overset{H_1}{\geq}} \left(\frac{I_0^{-1}(T_1)}{2|\alpha|^2} \right)^2 = T_2 \quad (1.39)$$

Finally, a normalization with respect to the output noise power, yields the following optimum group invariant test with threshold T_{mn} for a change of scale and starting phase angle ϕ (see [26]):

$$r_{mn} = \frac{\left| \mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1} \mathbf{x} \right|^2}{\mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1} \mathbf{v}(\omega_m, \vartheta_n)} \underset{H_0}{\overset{H_1}{\geq}} \frac{T_2}{\mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1} \mathbf{v}(\omega_m, \vartheta_n)} = T_{mn} \quad (1.40)$$

It is assumed for the present that \mathbf{R} is known except for a multiplicative scale factor. The test in (1.40) can be viewed in another way. Now let

$$\mathbf{R}^{-1/2} \mathbf{v}(\omega_m, \vartheta_n) = \tilde{\mathbf{v}}(\omega_m, \vartheta_n), \quad \frac{\tilde{\mathbf{v}}(\omega_m, \vartheta_n)}{\sqrt{\tilde{\mathbf{v}}^H(\omega_m, \vartheta_n) \tilde{\mathbf{v}}(\omega_m, \vartheta_n)}} = \bar{\mathbf{v}}(\omega_m, \vartheta_n) \quad \text{and} \quad \mathbf{R}^{-1/2} \mathbf{x} = \tilde{\mathbf{x}} \quad (1.41)$$

A substitution of the above transformations into (1.40) the test r_{mn} becomes

$$r_{mn} = \left| \bar{\mathbf{v}}^H(\omega_m, \vartheta_n) \tilde{\mathbf{x}} \right|^2 \underset{H_0}{\overset{H_1}{\geq}} T_{mn} \quad (1.42)$$

Thus, the test is equivalent to a measure of the projection of the whitened observation $\tilde{\mathbf{x}}$ onto the unit "whitened" steering vector $\bar{\mathbf{v}}(\omega_m, \vartheta_n)$.

The conditional pdf of the test r_{mn} under H_1 can be derived easily by the following theorem (see [29] and [37]):

Theorem 1.1 If \mathbf{x} is $N_m(\mathbf{m}_x, \mathbf{I}_m)$ and \mathbf{B} is an $m \times m$ projection matrix of rank k then $\mathbf{x}^H \mathbf{B} \mathbf{x}$ has a noncentral $\chi_k^2(\delta)$ distribution where $\delta = \mathbf{m}_x^H \mathbf{B} \mathbf{m}_x$.

In (1.42) $\tilde{\mathbf{x}}$ is $N_{MN}(\alpha \mathbf{R}^{-1/2} \mathbf{v}, 1)$ and $\bar{\mathbf{v}}(\omega_m, \vartheta_n) \bar{\mathbf{v}}^H(\omega_m, \vartheta_n)$ is a projection matrix of rank one. Then, by Theorem 1.1 the test r_{mn} is equivalent to $\chi_1^2(\rho_{mn})$, where ρ_{mn} is the generalized output signal-to-noise ratio of the detector given by

$$\rho_{mn} = |\alpha|^2 \mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1} \mathbf{v}(\omega_m, \vartheta_n) . \quad (1.43)$$

Hence the conditional pdf of r_{mn} under H_1 is given by

$$P(r_{mn}|H_1) = e^{-r_{mn}-\rho_{mn}} I_0(2\sqrt{\rho_{mn}r_{mn}}) U(r_{mn}) , \quad (1.44)$$

where $U(\cdot)$ represents the unit step function. By setting ρ_{mn} equal to zero in (1.44), the conditional pdf of r_{mn} under H_0 is obtained as

$$P(r_{mn}|H_0) = e^{-r_{mn}} U(r_{mn}) . \quad (1.45)$$

For the situation of a known steering vector and known covariance, the false-alarm probability, denoted by $P_{FA_{mn}}$, and the probability of detection, denoted by $P_{D_{mn}}$, are respectively given by

$$P_{FA_{mn}} = P_r\{r_{mn} > T_{mn}|H_0\} = \int_{T_{mn}}^{\infty} e^{-r_{mn}} dr_{mn} = e^{-T_{mn}} , \quad (1.46)$$

$$P_{D_{mn}} = \int_{T_{mn}}^{\infty} P(r_{mn}|H_1) dr_{mn} = \int_{T_{mn}}^{\infty} e^{-r_{mn}-\rho_{mn}} I_0(2\sqrt{\rho_{mn}r_{mn}}) dr_{mn} , \quad (1.47)$$

where the threshold T_{mn} for a given false alarm probability, is given by

$$T_{mn} = -\ln(P_{FA_{mn}}) . \quad (1.48)$$

Actually, the target steering vector $\mathbf{v}(\omega_m, \vartheta_n)$ is unknown in the real radar problem. For a single target, the standard technique is to evaluate the above test statistic over a discrete set of Doppler-angle vectors, i.e. forming a Doppler-angle filter bank, and declaring target presence if the maximum filter output of the filter bank exceeds the threshold T_h . Thus, the test r over the filter bank is given by

$$r = \max_{m,n} \{r_{mn}\} = \max_{m,n} \left\{ \frac{|\mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1} \mathbf{x}|^2}{\mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1} \mathbf{v}(\omega_m, \vartheta_n)} \right\} \underset{H_0}{\overset{H_1}{>}} T_h . \quad (1.49)$$

It is shown in the next section that this maximization over the filter bank also provides a ML estimate of the target steering vector. That means that the detection and estimation of Doppler and angle are performed simultaneously.

For the Doppler-angle filter bank case, by the independence of each Doppler-angle channel, the false-alarm probability, P_{FA} , and the overall threshold, T_h are related by the following formulae:

$$\begin{aligned} P_{FA} &= \Pr\left\{ \max_{\substack{1 \leq m \leq M \\ 1 \leq n \leq N}} (r_{mn}) > T_h | H_0 \right\} = 1 - \Pr\{r_{11} \leq T_h, r_{12} \leq T_h, \dots, r_{MN} \leq T_h | H_0\} \\ &= 1 - \prod_{n=1}^N \prod_{m=1}^M P_r\{r_{mn} \leq T_h | H_0\} = 1 - (1 - e^{-T_h})^{MN} , \end{aligned} \quad (1.50)$$

and

$$T_h = -\ln[1 - (1 - P_{FA})^{\frac{1}{MN}}] . \quad (1.51)$$

The probability of detection P_D for the Doppler-angle filter-bank is given approximately by

$$\begin{aligned}
P_D &= Pr\left\{ \underset{\substack{1 \leq m \leq M \\ 1 \leq n \leq N}}{Max}(r_{mn}) > T_h | H_1 \right\} = 1 - Pr\{r_{11} \leq T_h, r_{12} \leq T_h, \dots, r_{MN} \leq T_h | H_1\} \\
&= 1 - Pr\{r_{mn} \leq T_h | H_1\} \prod_{k=1}^{MN-1} Pr\{r_k \leq T_h | H_o\} \\
&= 1 - \int_0^{T_h} e^{-r_{mn} - \rho_{mn}} I_o(2\sqrt{\rho_{mn} r_{mn}}) dr_{mn} (1 - e^{-T_h})^{MN-1} \quad (1.52)
\end{aligned}$$

To arrive at this approximate, but quite accurate, result for large MN, it is assumed that the target can only appear in one Doppler-angle channel.

1.2.4 The Maximum Likelihood Estimator of Target Direction

Assume that the snapshots $\mathbf{x}(i)$ from different range gates i , for $i=1,2,\dots,K$, are independent. Then the joint pdf $P(\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(K))$ of the observed data from different range gates is given by

$$\begin{aligned}
P(\mathbf{x}(1), \dots, \mathbf{x}(K)) &= \prod_{i=1}^K P(\mathbf{x}(i)) \\
&= \left(\frac{1}{\pi}\right)^{MNK} |\mathbf{R}|^{-K} \exp\left(-\sum_{i=1}^K \mathbf{x}^H(i) \mathbf{R}^{-1} \mathbf{x}(i)\right) \quad (1.53)
\end{aligned}$$

Let $\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(K)]$. Then this joint pdf $P(\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(K))$ can be rewritten as follows :

$$\begin{aligned}
P(\mathbf{x}(1), \dots, \mathbf{x}(K)) &= \left(\frac{1}{\pi}\right)^{MNK} |\mathbf{R}|^{-K} \exp\left\{-\text{trace}\left(\mathbf{X}^H \mathbf{R}^{-1} \mathbf{X}\right)\right\} \\
&= \left[\left(\frac{1}{\pi}\right)^{MN} |\mathbf{R}|^{-1} \exp\left\{-\text{trace}\left(\mathbf{R}^{-1} \frac{1}{K} \mathbf{X} \mathbf{X}^H\right)\right\}\right]^K \quad (1.54)
\end{aligned}$$

The maximum likelihood estimate of the covariance $\hat{\mathbf{R}}$ is that value of \mathbf{R} which achieves the maximum,

$$\begin{aligned}
&\underset{\mathbf{R}}{Max} \left\{ \left[\left(\frac{1}{\pi}\right)^{MN} |\mathbf{R}|^{-1} \exp\left(-\text{trace}\left(\mathbf{R}^{-1} \frac{1}{K} \mathbf{X} \mathbf{X}^H\right)\right) \right]^K \right\} \\
&= \underset{\mathbf{R}}{Min} \left\{ \log |\mathbf{R}| + \text{trace}\left(\mathbf{R}^{-1} \mathbf{S}\right) \right\} \quad (1.55)
\end{aligned}$$

where $\mathbf{S} = \frac{1}{K} \mathbf{X} \mathbf{X}^H$. By taking the partial derivatives of (1.55) with respect to the components of \mathbf{R} and setting the result to zero, yields

$$\frac{\partial}{\partial \mathbf{R}} \left[\log |\mathbf{R}| + \text{trace}\left(\mathbf{R}^{-1} \mathbf{S}\right) \right] = 0 \quad (1.56)$$

or

$$\left(\mathbf{R}^{-1}\right)^T - \left(\mathbf{R}^{-1} \mathbf{S} \mathbf{R}^{-1}\right)^T = 0 \quad (1.57)$$

so that finally the ML estimate is given by

$$\hat{\mathbf{R}} = \mathbf{S}, \quad (1.58)$$

where $\frac{\partial}{\partial \mathbf{R}}$ denotes the partial derivatives with respect to the components of \mathbf{R}

Next, the maximum likelihood (ML) estimate of the complex amplitude α and the steering vector $\mathbf{v}(\omega_m, \vartheta_n)$ are derived by a use of the ML estimate $\hat{\mathbf{R}}$ in (1.58). The required ML estimate of the parameters, α and $\mathbf{v}(\omega_m, \vartheta_n)$, are defined by the pair $(\hat{\alpha}, \hat{\mathbf{v}}(\omega_m, \vartheta_n))$. This estimate is that pair $(\alpha, \mathbf{v}(\omega_m, \vartheta_n))$ which attains the maximum in the following maximization process :

$$\begin{aligned} \underset{\alpha, \mathbf{v}(\omega_m, \vartheta_n)}{\text{Max}} P(\mathbf{x}|\alpha, \mathbf{v}(\omega_m, \vartheta_n), \hat{\mathbf{R}}) &= \underset{\alpha, \mathbf{v}(\omega_m, \vartheta_n)}{\text{Max}} \log P(\mathbf{x}|\alpha, \mathbf{v}(\omega_m, \vartheta_n), \hat{\mathbf{R}}) \\ &= \underset{\alpha, \mathbf{v}(\omega_m, \vartheta_n)}{\text{Max}} \left\{ [\mathbf{x} - \alpha \mathbf{v}(\omega_m, \vartheta_n)]^H \hat{\mathbf{R}}^{-1} [\mathbf{x} - \alpha \mathbf{v}(\omega_m, \vartheta_n)] \right\} \\ &= \underset{\alpha, \mathbf{v}(\omega_m, \vartheta_n)}{\text{Max}} g(\alpha, \mathbf{v}(\omega_m, \vartheta_n)) \quad , \end{aligned} \quad (1.59)$$

where $g(\alpha, \mathbf{v}(\omega_m, \vartheta_n))$ is the log-likelihood function (excluding the constant terms). This maximization is performed first over α , as in [33] and [17], to obtain that value of α for which

$$\frac{\partial}{\partial \alpha} g(\alpha, \mathbf{v}(\omega_m, \vartheta_n)) = 0 \quad , \quad (1.60)$$

or

$$\begin{aligned} -\frac{\partial}{\partial \alpha} \left[\mathbf{x}^H \hat{\mathbf{R}}^{-1} \mathbf{x} - \alpha \mathbf{x}^H \hat{\mathbf{R}}^{-1} \mathbf{v}(\omega_m, \vartheta_n) - \alpha^* \mathbf{v}^H(\omega_m, \vartheta_n) \hat{\mathbf{R}}^{-1} \mathbf{x} \right. \\ \left. + \alpha^* \alpha \mathbf{v}^H(\omega_m, \vartheta_n) \hat{\mathbf{R}}^{-1} \mathbf{v}(\omega_m, \vartheta_n) \right] = 0 \quad , \end{aligned} \quad (1.61)$$

or

$$\mathbf{x}^H \hat{\mathbf{R}}^{-1} \mathbf{v}(\omega_m, \vartheta_n) - \alpha^* \mathbf{v}^H(\omega_m, \vartheta_n) \hat{\mathbf{R}}^{-1} \mathbf{v}(\omega_m, \vartheta_n) = 0 \quad . \quad (1.62)$$

Hence one obtains

$$\hat{\alpha} = \frac{\mathbf{v}^H(\omega_m, \vartheta_n) \hat{\mathbf{R}}^{-1} \mathbf{x}}{\mathbf{v}^H(\omega_m, \vartheta_n) \hat{\mathbf{R}}^{-1} \mathbf{v}(\omega_m, \vartheta_n)} \quad . \quad (1.63)$$

A substitution of the above result for $\hat{\alpha}$, the estimate in (1.63) of the amplitude α , into the log-likelihood function in (1.59) yields finally the ML estimate $\hat{\mathbf{v}}(\omega_m, \vartheta_n)$ which is that values of $\mathbf{v}(\omega_m, \vartheta_n)$ which achieves

$$\underset{\mathbf{v}(\omega_m, \vartheta_n)}{\text{Max}} \left\{ \frac{|\mathbf{v}^H(\omega_m, \vartheta_n) \hat{\mathbf{R}}^{-1} \mathbf{x}|^2}{\mathbf{v}^H(\omega_m, \vartheta_n) \hat{\mathbf{R}}^{-1} \mathbf{v}(\omega_m, \vartheta_n)} \right\} \quad . \quad (1.64)$$

1.2.5 Performance Analysis of the Detector (when the Noise Covariance is Unknown)

In the real radar problem, the noise covariance is unknown. The best way to find the required maximum likelihood ratio test for detection is to replace the noise covariance matrix with its

maximum likelihood estimate using the observed data for a set of different range gates in the vicinity of the range gate of the target to be detected. The test statistic under the signal-plus-noise hypothesis H_1 is derived first. If \mathbf{R} is replaced by $\hat{\mathbf{R}}$ in (1.40) and $\frac{1}{K}(\mathbf{X}\mathbf{X}^H)$ is substituted for $\hat{\mathbf{R}}$, the test r_{mn} becomes

$$\begin{aligned} r_{mn} &= \frac{|\mathbf{v}^H(\omega_m, \vartheta_n) \hat{\mathbf{R}}^{-1} \mathbf{x}|^2}{\mathbf{v}^H(\omega_m, \vartheta_n) \hat{\mathbf{R}}^{-1} \mathbf{v}(\omega_m, \vartheta_n)} \underset{H_0}{\overset{H_1}{>}} T_{mn} \\ &= (K) \frac{|\mathbf{v}^H(\omega_m, \vartheta_n) (\mathbf{X}\mathbf{X}^H)^{-1} \mathbf{x}|^2}{\mathbf{v}^H(\omega_m, \vartheta_n) (\mathbf{X}\mathbf{X}^H)^{-1} \mathbf{v}(\omega_m, \vartheta_n)} \underset{H_0}{\overset{H_1}{>}} T_{mn} . \end{aligned} \quad (1.65)$$

It is assumed that the observed data from the different range gates are independent and have the same statistics. Then a whitened \mathbf{X} is obtained as follows :

$$\mathbf{X} = [\mathbf{x}(1), \mathbf{x}(2), \dots, \mathbf{x}(K)] = [\mathbf{R}^{1/2} \mathbf{w}(1), \mathbf{R}^{1/2} \mathbf{w}(2), \dots, \mathbf{R}^{1/2} \mathbf{w}(K)] = \mathbf{R}^{1/2} \mathbf{W} , \quad (1.66)$$

where $\mathbf{W} = [\mathbf{w}(1), \mathbf{w}(2), \dots, \mathbf{w}(K)]$ is an $MN \times K$ matrix comprised of independently identically distributed (i.i.d.) complex Gaussian random variables with a zero mean and a unit variance, each random variable $[\mathbf{w}(l)]_i$, for $i=1, \dots, MN$, $l=1, \dots, K$, is distributed $N(0,1)$. By substituting $\mathbf{R}^{1/2} \mathbf{W}$ for \mathbf{X} , the test r_{mn} now becomes

$$r_{mn} = \frac{|\mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1/2} (\mathbf{W}\mathbf{W}^H)^{-1} \mathbf{R}^{-1/2} \mathbf{x}|^2}{\mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1/2} (\mathbf{W}\mathbf{W}^H)^{-1} \mathbf{R}^{-1/2} \mathbf{a}(\vartheta_s)} \underset{H_0}{\overset{H_1}{>}} \frac{T_{mn}}{K} = T'_{mn} . \quad (1.67)$$

Let \mathbf{Q} be an arbitrary unitary and Hermitian matrix and \mathbf{w} be a white Gaussian random vector with $\text{Cov}(\mathbf{w}) = I_{MN}$. The mean and covariance of $\mathbf{Q}\mathbf{w}$ is the same as that of \mathbf{w} . Thus, $\mathbf{Q}\mathbf{w}$ and \mathbf{w} are equivalent statistically since they are both Gaussian and have the same second order statistics. It follows that

$$\mathbf{Q}\mathbf{W} \underset{d}{=} \mathbf{W} \text{ and } \mathbf{W}\mathbf{W}^H \underset{d}{=} \mathbf{Q}\mathbf{W}\mathbf{W}^H\mathbf{Q} , \quad (1.68)$$

where " $\underset{d}{=}$ " denotes equality in distribution. Then r_{mn} in (1.67) becomes

$$r_{mn} \underset{d}{=} \frac{|\mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1/2} \mathbf{Q} (\mathbf{W}\mathbf{W}^H)^{-1} \mathbf{Q} \mathbf{R}^{-1/2} \mathbf{x}|^2}{\mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1/2} \mathbf{Q} (\mathbf{W}\mathbf{W}^H)^{-1} \mathbf{Q} \mathbf{R}^{-1/2} \mathbf{v}(\omega_m, \vartheta_n)} \underset{H_0}{\overset{H_1}{>}} T'_{mn} . \quad (1.69)$$

By the Householder transformation (see[35],[34] and Appendix A) one can find next a unitary matrix \mathbf{Q} such that

$$\mathbf{Q}\mathbf{R}_m^{-1/2} \mathbf{v}(\omega_m, \vartheta_n) = \sqrt{\mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}_m^{-1} \mathbf{v}(\omega_m, \vartheta_n)} \mathbf{e}_1 e^{j\theta_1} , \quad (1.70)$$

where $\mathbf{e}_1 = [1, 0, \dots, 0]^T$ is a unit vector, $\theta_1 = \pi + \arg([\mathbf{R}_m^{-1/2} \mathbf{a}(\vartheta_s)]_1)$. $[\mathbf{z}]_1$ = first coordinate component of \mathbf{z} , and $\arg(x + jy) = \tan^{-1}(y/x)$. By the above result one obtains the following equivalent test :

$$r_{mn} = \frac{|\sqrt{\mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}_m^{-1} \mathbf{a}(\vartheta_s)} \mathbf{e}_1^H (\mathbf{W}\mathbf{W}^H)^{-1} \mathbf{Q} \mathbf{R}^{-1} \mathbf{x}|^2}{(\mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}_m^{-1} \mathbf{v}(\omega_m, \vartheta_n)) \mathbf{e}_1^H (\mathbf{W}\mathbf{W}^H)^{-1} \mathbf{e}_1} \underset{H_0}{\overset{H_1}{>}} T'_{mn} , \quad (1.71)$$

or

$$r_{mn} = \frac{|\mathbf{e}_1^H (\mathbf{W}\mathbf{W}^H)^{-1} \mathbf{Q}\mathbf{R}^{-1/2} \mathbf{x}|^2}{[(\mathbf{W}\mathbf{W}^H)^{-1}]_{1,1}} \underset{H_o}{\overset{H_1}{>}} T'_{mn} , \quad (1.72)$$

where $[(\mathbf{W}\mathbf{W}^H)^{-1}]_{1,1}$ represents the (1,1) scalar element of the matrix $(\mathbf{W}\mathbf{W}^H)^{-1}$. In the above equation, $\mathbf{Q}\mathbf{R}^{-1/2} \mathbf{x}$ can be simplified in the following manner:

$$\begin{aligned} \mathbf{Q}\mathbf{R}^{-1/2} \mathbf{x} &= \mathbf{Q}\mathbf{R}^{-1/2} (\alpha \mathbf{v}(\omega_m, \vartheta_n) + \mathbf{n}) \\ &= \exp \left\{ j \left(\pi + \arg \left([\mathbf{R}^{-1/2} \mathbf{v}(\omega_m, \vartheta_n)]_1 \right) \right) \right\} \alpha \sqrt{\mathbf{v}^H(\omega_m, \vartheta_n) \mathbf{R}^{-1} \mathbf{v}(\omega_m, \vartheta_n)} \mathbf{e}_1 + \mathbf{Q}\mathbf{w} \\ &\stackrel{d}{=} \mathbf{u} \end{aligned} \quad (1.73)$$

where \mathbf{w} is a $MN \times 1$ white Gaussian random vector and \mathbf{u} is a $MN \times 1$ vector with independent entries, $N(0,1)$, except that the first component, u_1 , which is $N(m'_{Z_{mn}}, 1)$ with $m'_{Z_{mn}} = \sqrt{\rho_{mn}} \exp \left\{ j \left(\pi + \arg \left([\mathbf{R}^{-1/2} \mathbf{v}(\omega_m, \vartheta_n)]_1 \right) \right) \right\}$. A substitution of (1.73) into (1.72) yields the test,

$$r_{mn} \stackrel{d}{=} \frac{|\mathbf{e}_1^H (\mathbf{W}\mathbf{W}^H)^{-1} \mathbf{u}|^2}{[(\mathbf{W}\mathbf{W}^H)^{-1}]_{1,1}} \underset{H_o}{\overset{H_1}{>}} T'_{mn} . \quad (1.74)$$

Before deriving the pdf of the one-filter test r_{mn} in (1.74), the pdf of $[(\mathbf{W}\mathbf{W}^H)^{-1}]_{1,1}$ is derived first. Let \mathbf{u} be a $MN \times 1$ vector which is partitioned as follows:

$$\mathbf{u} = \begin{bmatrix} u_1 \\ \mathbf{u}' \end{bmatrix} \quad (1.75)$$

where u_1 is a scalar and \mathbf{u}' is a $(MN-1) \times 1$ vector. Next, \mathbf{W} is partitioned in the manner,

$$\mathbf{W} = \begin{bmatrix} \mathbf{z}_1^H \\ \mathbf{Z}^H \end{bmatrix}, \quad (1.76)$$

where \mathbf{z}_1^H is a $1 \times K$ vector and \mathbf{Z}^H is a $(MN-1) \times K$ matrix. From (1.76) one obtains the following partition for $\mathbf{W}\mathbf{W}^H$:

$$\mathbf{W}\mathbf{W}^H = \begin{bmatrix} \mathbf{z}_1^H \mathbf{z}_1 & \mathbf{z}_1^H \mathbf{Z} \\ \mathbf{Z}^H \mathbf{z}_1 & \mathbf{Z}^H \mathbf{Z} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}. \quad (1.77)$$

From [37], the inverse $(\mathbf{W}\mathbf{W}^H)^{-1}$ of the partitioned matrix $\mathbf{W}\mathbf{W}^H$ is given by the following formula:

$$(\mathbf{W}\mathbf{W}^H)^{-1} = \begin{bmatrix} [(\mathbf{W}\mathbf{W}^H)^{-1}]_{1,1} & [(\mathbf{W}\mathbf{W}^H)^{-1}]_{1,2} \\ [(\mathbf{W}\mathbf{W}^H)^{-1}]_{2,1} & [(\mathbf{W}\mathbf{W}^H)^{-1}]_{2,2} \end{bmatrix} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix}, \quad (1.78)$$

where

$$\begin{aligned} \mathbf{B}_{11} &= (\mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{21})^{-1}, \\ \mathbf{B}_{21} &= -\mathbf{A}_{22}^{-1} \mathbf{A}_{21} (\mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{21})^{-1}, \\ \mathbf{B}_{12} &= -(\mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{21})^{-1} \mathbf{A}_{12} \mathbf{A}_{22}^{-1}, \\ \mathbf{B}_{22} &= \mathbf{A}_{22}^{-1} \mathbf{A}_{21} (\mathbf{A}_{11} - \mathbf{A}_{12} \mathbf{A}_{22}^{-1} \mathbf{A}_{21})^{-1} \mathbf{A}_{12} \mathbf{A}_{22}^{-1} + \mathbf{A}_{22}^{-1}. \end{aligned} \quad (1.79)$$

Then, by the above formula $\eta = [\mathbf{W}\mathbf{W}^{-1}]_{1,1}$ is obtained as

$$\eta = \left(\mathbf{z}_1^H \mathbf{z}_1 - \mathbf{z}_1^H \mathbf{Z} (\mathbf{Z}^H \mathbf{Z})^{-1} \mathbf{Z}^H \mathbf{z}_1 \right)^{-1} \stackrel{d}{=} \frac{1}{\chi_{K-N+1}^2}, \quad (1.80)$$

(see Appendix B), where χ_m^2 is a complex normalized chi-squared random variable with m degrees of freedom. A substitution of (1.79) into (1.78) and then along with (1.80) into (1.74), yields the following result for r_{mn} :

$$r_{mn} = \frac{\left| u_1 [\mathbf{W}\mathbf{W}^{-1}]_{1,1} + [\mathbf{W}\mathbf{W}^{-1}]_{1,2} \mathbf{u}' \right|^2}{[\mathbf{W}\mathbf{W}^{-1}]_{1,1}} = \eta \left| u_1 - \mathbf{z}_1^H \mathbf{Z} (\mathbf{Z}^H \mathbf{Z})^{-1} \mathbf{u}' \right|^2. \quad (1.81)$$

Since the summation of the independent Gaussian random variables is also a Gaussian random variable, one obtains

$$\begin{aligned} u_1 - \mathbf{z}_1^H \mathbf{Z} (\mathbf{Z}^H \mathbf{Z})^{-1} \mathbf{u}' &\stackrel{d}{=} N(m'_{Z_{mn}}, 1) - (\mathbf{z}_1^H \mathbf{Z} (\mathbf{Z}^H \mathbf{Z})^{-2} \mathbf{Z}^H \mathbf{z}_1)^{1/2} N(0, 1) \\ &\stackrel{d}{=} \left(1 + \mathbf{z}_1^H \mathbf{Z} (\mathbf{Z}^H \mathbf{Z})^{-2} \mathbf{Z}^H \mathbf{z}_1 \right)^{1/2} \tilde{u}, \end{aligned} \quad (1.82)$$

where \tilde{u} is $N(m'_{Z_{mn}}, 1)$. Thus r_{mn} in (1.81) becomes

$$r_{mn} \stackrel{d}{=} \frac{\eta}{\zeta} |\tilde{u}|^2. \quad (1.83)$$

Here ζ is equivalent to a beta distributed variate (see Appendix B) defined as follows:

$$\zeta = \left(1 + \mathbf{z}_1^H \mathbf{Z} (\mathbf{Z}^H \mathbf{Z})^{-2} \mathbf{Z}^H \mathbf{z}_1 \right)^{-1}. \quad (1.84)$$

The pdf of this beta distribution is given by

$$P(\zeta) = \frac{1}{B(MN-1, K-MN+2)} \zeta^{K-MN+1} (1-\zeta)^{MN-2}, \quad (1.85)$$

where $B(M, N)$ is the beta function defined by

$$B(M, N) = \frac{(M-1)!(N-1)!}{(M+N-1)!}. \quad (1.86)$$

Since the magnitude squared of a nonzero mean, say m , Gaussian random variable is equivalent to a non-central $\chi_1^2(m^H m)$ variate, one has $|\tilde{u}|^2 \stackrel{d}{=} \chi_1^2(\rho_{mn})$. This test r_{mn} in (1.83) finally becomes

$$r_{mn} \stackrel{d}{=} \frac{\chi_1^2(\rho_{mn})}{\chi_{K-MN+1}^2 \zeta}. \quad (1.87)$$

Since $(n_2 \chi_{n_1}^2(\rho_{mn})) / (n_1 \chi_{n_2}^2)$ is a noncentral F distribution (see [31], [29] and [37]), the conditional pdf of r_{mn} , under ζ and H_1 , is given by

$$P(r_{mn} | \zeta, H_1) = \frac{e^{-\rho_{mn}}}{B(1, K-MN+1)} \frac{\zeta}{(1+r_{mn}\zeta)^{K-MN+2}} {}_1F_1(K-MN+2, 1, \frac{\rho_{mn}\zeta r_{mn}}{1+\zeta r_{mn}}) \quad (1.88)$$

where ${}_1F_1(a; b; x)$ is the confluent hypergeometric function defined by

$${}_1F_1(a; b; x) = \sum_{k=0}^{\infty} \frac{a(a+1) \cdots (a+k-1)}{b(b+1) \cdots (b+k-1)} \frac{x^k}{k!}. \quad (1.89)$$

Averaging $P(r_{mn}|\zeta, H_1)$ over ζ , one has the conditional pdf of r_{mn} , under H_1 , as follows:

$$P(r_{mn}|H_1) = \frac{1}{B(1, K - MN + 1)B(MN - 1, K - MN + 2)} \times \int_0^1 \frac{e^{-\rho_{mn}\zeta} \zeta^{K-MN+2} (1-\zeta)^{MN-2}}{(1+r_{mn}\zeta)^{K-MN+2}} {}_1F_1(K - MN + 2, 1, \frac{\rho_{mn}\zeta r_{mn}}{1+\zeta r_{mn}}) d\zeta. \quad (1.90)$$

Setting $\rho_{mn} = 0$ in the above equation, the conditional pdf of r_{mn} , under H_0 , is given by

$$P(r_{mn}|H_0) = \frac{1}{B(1, K - MN + 1)B(MN - 1, K - MN + 2)} \int_0^1 \frac{\zeta^{K-MN+2} (1-\zeta)^{MN-2}}{(1+r_{mn}\zeta)^{K-MN+2}} d\zeta. \quad (1.91)$$

From (1.46) and (1.91) the false-alarm probability $\hat{P}_{FA_{mn}}$ of a single Doppler-angle filter is given finally by

$$\begin{aligned} \hat{P}_{FA_{mn}} &= \frac{1}{B(1, K - MN + 1)B(MN - 1, K - MN + 2)} \times \\ &\quad \int_{T'_{mn}}^{\infty} \int_0^1 \frac{\zeta^{K-MN+2} (1-\zeta)^{MN-2}}{(1+r_{mn}\zeta)^{K-MN+2}} d\zeta dr_{mn} \\ &= \frac{1}{B(MN - 1, K - MN + 2)} \int_0^1 \left(\frac{\zeta}{1+T'_{mn}\zeta} \right)^{K-MN+1} (1-\zeta)^{MN-2} d\zeta. \quad (1.92) \end{aligned}$$

From (1.50) and (1.91) the false-alarm probability of the Doppler-angle filter bank, \hat{P}_{FA} , is given by

$$\hat{P}_{FA} = 1 - \left(\frac{1}{B(1, K - MN + 1)B(MN - 1, K - MN + 2)} \times \int_0^{T'_h} \int_0^1 \frac{\zeta^{K-MN+2} (1-\zeta)^{MN-2}}{(1+r_{mn}\zeta)^{K-MN+2}} d\zeta dr_{mn} \right)^{MN}. \quad (1.93)$$

Note that the above expressions for $\hat{P}_{FA_{mn}}$ and \hat{P}_{FA} are functions only of K (the number of range gates used to estimate the covariance), N (the number of antenna elements), and M (the number of pulses in CPI). Hence these false-alarm probabilities are independent of the signal and noise power. As a consequence, the test in (1.65) is a CFAR criterion.

The probability of detection for the single Doppler-angle channel case is obtained from (1.47) and (1.90) as follows :

$$\begin{aligned} \hat{P}_{D_{mn}} &= \frac{1}{B(1, K - MN + 1)B(MN - 1, K - MN + 2)} \times \\ &\quad \int_{T'_{mn}}^{\infty} \int_0^1 \frac{e^{-\rho_{mn}\zeta} \zeta^{K-MN+2} (1-\zeta)^{MN-2}}{(1+r_{mn}\zeta)^{K-MN+2}} {}_1F_1(K - MN + 2, 1, \frac{\rho_{mn}\zeta r_{mn}}{1+\zeta r_{mn}}) d\zeta dr_{mn} \\ &= 1 - \frac{e^{-\rho_{mn}}}{B(MN - 1, K - MN + 2)} \times \\ &\quad \int_0^1 \zeta^{K-MN+1} (1-\zeta)^{MN-2} \sum_{k=0}^{\infty} \frac{\rho_{mn}^k}{k!} I_{\frac{\zeta T'_{mn}}{1+\zeta T'_{mn}}} (k+1, K - MN + 1) d\zeta, \quad (1.94) \end{aligned}$$

where $I_x(a, b)$ is an incomplete beta function.

Similarly, the probability of detection for the Doppler-angle filter matrix case can be obtained finally from (1.52), (1.90) and (1.91) as follows:

$$\begin{aligned} \hat{P}_D = & 1 - \frac{1}{B(1, K - MN + 1)B(MN - 1, K - MN + 2)} \times \\ & \int_0^{T'_h} \int_0^1 \frac{e^{-\rho_{mn}\zeta} \zeta^{K-MN+2} (1-\zeta)^{MN-2}}{(1+r_{mn}\zeta)^{K-MN+2}} {}_1F_1(K - MN + 2, 1, \frac{\rho_{mn}\zeta r_{mn}}{1+\zeta r_{mn}}) d\zeta dr_{mn} \times \\ & \left(\frac{1}{B(1, K - MN + 1)B(MN - 1, K - MN + 2)} \int_0^{T'_h} \int_0^1 \frac{\zeta^{K-MN+2} (1-\zeta)^{MN-2}}{(1+r_{mn}\zeta)^{K-MN+2}} d\zeta dr_{mn} \right)^{MN-1} \end{aligned} \quad (1.95)$$

The probability of detection as a function of the generalized signal-to-noise ratio, ρ_{mn} , is shown in Figures 1.13 and 1.14 for $M=N=10$ and $M=N=5$, respectively. This leads to the two kinds of curves, plotted on the same figures, one for a single channel and another for a filter bank. With the same parameters (M , N and K), the performance of the single channel is better than that of the filter bank. From Figures 1.13 and 1.14, one can find the effect of increasing the numbers M and N in the filter bank on the performance. As the number of filters in the filter bank becomes larger, the performance of the filter bank is much degraded from that of the single channel. The effect of sample size K on the performance is not very significant for $M=N=10$ in Figure 1.13, but about 2 dB of degradation as K goes from $5MN$ to $2MN$ as shown on Figure 1.14 for $M=N=5$.

1.2.6 Performance Analyses of Two Partially Adaptive STAP Detectors

Two types of partially adaptive STAPs, element-space post-Doppler STAP and beam-space pre-Doppler STAP, are discussed in this section. Both of these criteria have the advantage of reduced dimension but both sacrifice some performance. The element-space post-Doppler STAP uses non-adaptive temporal filtering first and then performs the spatial filtering adaptively. In contrast, the beam-space pre-Doppler STAP performs in the reverse order, i.e. non-adaptive spatial filtering first, followed by adaptive temporal filtering.

CASE 1: Element-Space Post-Doppler STAP

Let \mathbf{y}_m be the $N \times 1$ output vector of the m^{th} Doppler filter. Then a snapshot of data in the presence of a target \mathbf{x} within the m^{th} Doppler filter is expressed by

$$\mathbf{y}_m = (\mathbf{f}_m \otimes \mathbf{I}_N)^H \mathbf{x} = (\mathbf{f}_m^H \otimes \mathbf{I}_N)(\mathbf{s} + \mathbf{n}) = [\alpha \mathbf{f}_m^H \mathbf{b}(\omega_m)] \mathbf{a}(\vartheta_n) + (\mathbf{f}_m \otimes \mathbf{I}_N)^H \mathbf{n} \quad , \quad (1.96)$$

where \mathbf{f}_m represents a $M \times 1$ DFT vector for the m^{th} Doppler filter.

Since the DFT is a set of M linear operations, one has $\mathbf{n}_m = (\mathbf{f}_m \otimes \mathbf{I}_N)^H \mathbf{n}$ so that the output noise of the m^{th} filter is a Gaussian random vector. The Doppler channel output noise vectors \mathbf{n}_m for $m=1, 2, \dots, M$ are statistically independent [38] by the assumption that the correlation structure of the noise is much less than the duration of the observations. The detection test with a known covariance matrix for this element-space post-Doppler STAP is given by

$$r = \max_{m,n} \left\{ \frac{|\mathbf{a}^H(\vartheta_n) \mathbf{R}_m^{-1} \mathbf{y}_m|^2}{\mathbf{a}^H(\vartheta_n) \mathbf{R}_m^{-1} \mathbf{a}(\vartheta_n)} \right\} \underset{H_0}{\overset{H_1}{>}} T_{mn} \quad , \quad (1.97)$$

where $\mathbf{R}_m = E\{\mathbf{n}_m \mathbf{n}_m^H\}$ is the m^{th} Doppler channel output noise covariance. The maximum of this test is obtained over all beams for each Doppler filter. This STAP detector suppresses the interference and noise by adaptively placing nulls in the angular domain only, and using DFT filtering in the Doppler frequency domain.

The ML estimate $\hat{\mathbf{a}}(\vartheta_n)$ of the target angle for a given Doppler frequency, ω_m , is obtained by the same argument used in the previous section for estimation. That is, $\hat{\mathbf{a}}(\vartheta_n)$ is that value of $\mathbf{a}(\vartheta_n)$ which obtains the maximum,

$$\max_{\mathbf{a}(\vartheta_n)} \left\{ \frac{|\mathbf{a}^H(\vartheta_n) \hat{\mathbf{R}}_m^{-1} \mathbf{y}_m|^2}{\mathbf{a}^H(\vartheta_n) \hat{\mathbf{R}}_m^{-1} \mathbf{a}(\vartheta_n)} \right\} \quad (1.98)$$

Note that this STAP detector provides simultaneous detection and ML estimation for the target angle only.

Similarly, the false-alarm probability for the single Doppler channel, $P_{FA_{mn}}$, and for the Doppler-filter bank, P_{FA} , are given respectively by

$$P_{FA_{mn}} = e^{-T_{mn}} \quad (1.99)$$

$$P_{FA} = 1 - (1 - e^{-T_h})^{MN} \quad (1.100)$$

The probability of detection for a single-Doppler channel, $P_{D_{mn}}$, and for the Doppler-filter bank, P_D , are given, respectively, by

$$P_{D_{mn}} = \int_{T_{mn}}^{\infty} e^{-r_{mn} - \tilde{\rho}_{mn}} I_0(2\sqrt{\tilde{\rho}_{mn} r_{mn}}) dr_{mn} \quad (1.101)$$

$$P_D = 1 - \int_0^{T_h} e^{-r_{mn} - \tilde{\rho}_{mn}} I_0(2\sqrt{\tilde{\rho}_{mn} r_{mn}}) dr_{mn} (1 - e^{-T_h})^{MN-1} \quad (1.102)$$

where the generalized signal-to-noise ratio $\tilde{\rho}_{mn}$ is given by

$$\tilde{\rho}_{mn} = \left| \alpha \mathbf{f}_m^H \mathbf{b}(\omega_m) \right|^2 \mathbf{a}^H(\vartheta_n) \mathbf{R}_m^{-1} \mathbf{a}(\vartheta_n) \quad (1.103)$$

When the noise covariance \mathbf{R}_m of the m^{th} Doppler filter output is unknown, \mathbf{R}_m in (1.97) is replaced by its ML estimate $\hat{\mathbf{R}}_m$ which is given by

$$\hat{\mathbf{R}}_m = \frac{1}{K} \sum_{k=1}^K \mathbf{y}_m(k) \mathbf{y}_m^H(k) \quad (1.104)$$

where $\mathbf{y}_m(k)$ represents the observed data at the m^{th} Doppler filter output from the nearby k^{th} range gate. Then, the false-alarm probability with unknown covariance for single-Doppler channel, $\hat{P}_{FA_{mn}}$, and for the Doppler-filter bank, \hat{P}_{FA} , are given, respectively, by

$$\hat{P}_{FA_{mn}} = \frac{1}{B(1, K - N + 1) B(N - 1, K - N + 2) (K - N + 1)} \times \int_0^1 \left(\frac{\zeta}{1 + T_{mn} \zeta} \right)^{K-N+1} (1 - \zeta)^{N-2} d\zeta \quad (1.105)$$

$$\hat{P}_{FA} = 1 - \left(\int_0^{T_h} \frac{1}{B(1, K - N + 1) B(N - 1, K - N + 2)} \int_0^1 \frac{\zeta^{K-N+2} (1 - \zeta)^{N-2}}{(1 + r_{mn} \zeta)^{K-N+2}} d\zeta dr_{mn} \right)^{MN} \quad (1.106)$$

The probability of detection $\hat{P}_{D_{mn}}$ with unknown covariance for a single Doppler channel and \hat{P}_D for the Doppler-filter bank are given, respectively, by

$$\hat{P}_{D_{mn}} = 1 - \frac{e^{-\tilde{\rho}_{mn}}}{B(N-1, K-N+2)} \int_0^1 \zeta^{K-N+1} (1-\zeta)^{N-2} \times \sum_{k=0}^{\infty} \frac{\tilde{\rho}_{mn}^k}{k!} I_{\frac{\zeta T'_{hm}}{1+\zeta T'_{hm}}} (k+1, K-N+1) d\zeta \quad (1.107)$$

$$\begin{aligned} \hat{P}_D = & 1 - \int_0^{T'_h} \frac{1}{B(1, K-N+1)B(N-1, K-N+2)} \times \\ & \int_0^1 \frac{e^{-\tilde{\rho}_{mn}\zeta} \zeta^{K-N+2} (1-\zeta)^{N-2}}{(1+r_{mn}\zeta)^{K-N+2}} {}_1F_1(K-N+2, 1, \frac{\tilde{\rho}_{mn}\zeta r_{mn}}{1+\zeta r_{mn}}) d\zeta dr_{mn} \times \\ & \left(\int_0^{T'_h} \frac{1}{B(1, K-N+1)B(N-1, K-N+2)} \int_0^1 \frac{\zeta^{K-N+2} (1-\zeta)^{N-2}}{(1+r_{mn}\zeta)^{K-N+2}} d\zeta dr_{mn} \right)^{MN-1} \end{aligned} \quad (1.108)$$

CASE 2: Beam-Space Pre-Doppler STAP

Let \mathbf{z}_n be the $M \times 1$ output vector of the n^{th} beam. Then a snapshot of data \mathbf{x} with a target present and \mathbf{z}_n are related by

$$\mathbf{z}_n = (\mathbf{I}_M \otimes \mathbf{g}_n)^H \mathbf{x} = (\mathbf{I}_M \otimes \mathbf{g}_n^H)(\mathbf{s} + \mathbf{n}) = [\alpha \mathbf{g}_n^H \mathbf{b}(\omega_m)] \mathbf{a}(\vartheta_n) + (\mathbf{I}_M \otimes \mathbf{g}_n)^H \mathbf{n} \quad , \quad (1.109)$$

where \mathbf{g}_n represents a $N \times 1$ spatial DFT vector for the n^{th} beam.

By the same arguments used in Case 1 the final detection test with a known covariance matrix for the beam-space pre-Doppler STAP is given by

$$r = \max_{m,n} \left\{ \frac{|\mathbf{b}^H(\omega_m) \mathbf{R}_n^{-1} \mathbf{z}_n|^2}{\mathbf{b}^H(\omega_m) \mathbf{R}_n^{-1} \mathbf{b}(\omega_m)} \right\} \frac{H_1}{H_0} T_{mn} \quad , \quad (1.110)$$

where \mathbf{R}_n represents the output noise covariance of the n^{th} beam. In contrast to Case 1 the beam-space pre-Doppler STAP suppresses the interference by placing nulls in the Doppler-frequency domain and in the angle domain by using fixed (non-adaptive) spatial DFT filtering only.

Similarly, the ML estimate $\hat{\mathbf{b}}(\omega_m)$ of Doppler frequency is that value of $\mathbf{b}(\omega_m)$ which obtains the maximum,

$$\max_{\mathbf{b}(\omega_m)} \left\{ \frac{|\mathbf{b}^H(\omega_m) \hat{\mathbf{R}}_n^{-1} \mathbf{z}_n|^2}{\mathbf{b}^H(\omega_m) \hat{\mathbf{R}}_n^{-1} \mathbf{b}(\omega_m)} \right\} \quad . \quad (1.111)$$

In contrast to Case 1, the detector for this case provides simultaneous adaptive detection and ML estimation of the Doppler frequency only.

All of the performance results of the test for the beam-space pre-Doppler STAP are obtained by simply exchanging N with M and substituting $\tilde{\rho}_{mn} = \left| \alpha \mathbf{g}_n^H \mathbf{a}(\vartheta_n) \right|^2 \mathbf{b}^H(\omega_m) \mathbf{R}_n^{-1} \mathbf{b}(\omega_m)$ for $\tilde{\rho}_{mn}$ in the corresponding generalized signal-to-noise ratio for the element-space post-Doppler STAP.

The probability of detection of these partially adaptive STAP's as a function of the generalized signal-to-noise ratio ρ_{mn} , is shown in Figures 1.15 and 1.16 for $M=N=10$ and $M=N=5$, respectively. The effect of sample size K on the performance is presented in these figures.

Performance comparisons between fully and partially adaptive STAP are made in Figures 1.17 and 1.18 for $M=N=10$ and $M=N=5$. The performance difference between partially and fully adaptive STAP is not large for a sample size K which is five times the number of degrees of freedom for $M=N=10$ and $M=N=5$. With sample size K which is twice the number of degrees of freedom the performance degradation of partial STAP with respect to full STAP processing is about 5 dB for $M=N=10$ and about 7 dB for $M=N=5$. Interestingly, the performance of the fully adaptive STAP with $M=N=5$ and $K=5MN$ is about the same as that of partially adaptive STAP with $M=N=5$ and $K=5N$.

1.2.7 Conclusion

A simultaneous CFAR detection and ML estimation algorithm is developed for both fully and partially adaptive STAP in this study. The performance of these CFAR tests are analyzed for a known and a maximum likelihood estimated covariance matrix. The performance curves of the probability of detection show that the performance is a function of the sample size K used to estimate the covariance. The choice of the number K is a tradeoff between performance and computational complexity. The optimum number is between two and five for K . The performance curves also show that the finer the resolution is, the worse the detection performance is expected to be.

1.3 A Fast CFAR Detection Space-Time Adaptive Processing Algorithm

All of the conventional CFAR detection algorithms which use space-time processing involve a time-consuming matrix inversion operation. Based on today's technology, this computational complexity sometimes makes the full-rank solution unrealizable. In this section, a CFAR detection algorithm, which does not need a matrix inversion, is developed by an adaptation and extension of Hotelling's principal component method studied recently [13] by Kirsteins and Tufts. If also the HT method [30] is incorporated into this new algorithm, the computational complexity is reduced to $O(N)$ from $O(N^3)$ for the matrix inversion used in the conventional adaptive algorithm.

1.3.1 Introduction

Future airborne radar are expected to be required to provide long-range detection of targets with increasingly smaller cross-sections. This function must be performed in environments where the clutter can be quite severe. Space-Time Adaptive Processing (STAP) refers to multidimensional adaptive filtering algorithms that simultaneously combine the signals from the elements of an array antenna and the multiple pulses of a coherent radar waveform to suppress interference and provide target detection.

The first published work on STAP for radar was by Brennan and Reed [17] in 1973, in which optimum space-time filtering was described. Later in 1986 a more all-encompassing generalized likelihood ratio test (GLRT) on STAP for radar detection was derived by Kelly [24]. Kelly used the sample matrix inverse (SMI) technique of Reed, Brennan and Mallett in [25]. Recently Chen and Reed [26] developed a simpler CFAR detection test than the GLRT obtained by Kelly. Robey, Fuhrmann, Kelly and Nitzberg [27] also developed independently the same result as Chen and

Reed. Although this new CFAR test showed a slightly degraded performance, it was easier to analyze than the GLRT. However, all of the previous STAP algorithms involved a time-consuming matrix inversion operation.

The principal components (PC) technique was developed originally by Hotelling [28]. Hotelling showed that the dimension of the problem often could be reduced without sacrificing too much of the information contained in the covariance matrix. Muirhead's book [29] is an excellent reference for the PC method. A new unnormalized GLRT which uses the PC technique is developed by Kirsteins and Tufts in [13], but they do not obtain the probability density functions (PDF) needed to evaluate the performance of the test statistic.

In this study the normalized LRT in [26] and [27] is modified by a use of the PC technique. This test exhibits the very desirable property that no matrix inversion is needed and that the computation complexity can be reduced to $O(N)$ by incorporating the HT method found recently by Hung and Turner in [30].

The PDF of this new detection statistic is derived here for both the noise-alone case and the signal-plus-noise case. In the noise-alone case, the PDF proves to be very simple and the probability of a false-alarm (PFA) is shown to depend only on the number of parameters and the sample data size. Thus this new PC test is a CFAR criterion.

1.3.2 Derivation of the Test Function

The system under consideration is a coherent pulsed Doppler radar. For simplicity the radar antenna is assumed to be a uniformly spaced linear array antenna, consisting of N_a elements. The radar transmits a coherent burst of N_p pulses. For each pulse repetition interval (PRI) N_r range samples are collected. With N_p pulses and N_a receiver channels, the received data for one coherent-processing interval (CPI) comprises $N_r N_p N_a$ complex baseband samples, the so-called $N_r N_p N_a$ data cube.

It is assumed in the present treatment that a target remains in only one range gate during a CPI. The snapshot of the $N = N_p N_a$ samples of data for the range gate, which is assumed to contain the target signal, is called the primary data. The secondary data consists of the outputs of L range gates in the vicinity of the range gate of the target to be detected.

The noise components of the data vectors consist of thermal noise, jamming or clutter, which are modeled as zero-mean, complex Gaussian, random noise vectors. The noise components of each range gate are assumed to share a common covariance matrix and to be mutually independent.

Let \mathbf{x} be a $N \times 1$ snapshot of a given range gate, i.e. the observation vector,

$$\mathbf{x} = [x_1, x_2, \dots, x_N]^T. \quad (1.112)$$

Two hypotheses are postulated :

$$H_0: \mathbf{x} = \mathbf{n}, \quad (1.113)$$

$$H_1: \mathbf{x} = \mathbf{s} + \mathbf{n}, \quad (1.114)$$

where \mathbf{n} is a $N \times 1$ Gaussian noise-plus-clutter vector with known covariance matrix \mathbf{R} and \mathbf{s} is a known $N \times 1$ target vector with a complex amplitude α and a random phase ϕ . Thus \mathbf{s} can be represented by the N -dimensional vector, $\mathbf{s} = \alpha \mathbf{v}$, where \mathbf{v} is a $N \times 1$ steering vector. Then the

usual decision rule, which is based on the matched filter for incoherent detection (see [26]), is given by

$$r = \left| \mathbf{v}^H \mathbf{R}^{-1} \mathbf{x} \right|^2 \underset{H_o}{\overset{H_1}{>}} T_o. \quad (1.115)$$

A normalization with respect to the output noise power yields a group-invariant test with respect to the groups of change-of-scale and starting phase as follows:

$$r = \frac{\left| \mathbf{v}^H \mathbf{R}^{-1} \mathbf{x} \right|^2}{\mathbf{v}^H \mathbf{R}^{-1} \mathbf{v}} \underset{H_o}{\overset{H_1}{>}} T'_o. \quad (1.116)$$

This test is independent of the scale of the assumed known covariance matrix \mathbf{R} . The generalized output signal-to-noise ratio, associated with the detection test in (1.116) is given by

$$\rho = \frac{E\{r|\mathbf{x} = \alpha \mathbf{v}\}}{E\{r|\mathbf{x} = \mathbf{n}\}} = |\alpha|^2 \mathbf{v}^H \mathbf{R}^{-1} \mathbf{v}, \quad (1.117)$$

where $E\{\}$ denotes the expectation-value operator.

1.3.3 Rank Reduction of the Clutter-Plus-Noise Covariance

The $N \times N$ covariance matrix \mathbf{R} of the background clutter and noise can be modeled as

$$\mathbf{R} = \mathbf{U} \mathbf{\Phi} \mathbf{U}^H + \sigma^2 \mathbf{I}, \quad (1.118)$$

where $\sigma^2 \mathbf{I}$ represents the covariance of the thermal noise and $\mathbf{U} \mathbf{\Phi} \mathbf{U}^H$ equals the covariance of the interference. Here \mathbf{U} is the $N \times N$ eigenvector matrix, and $\mathbf{\Phi}$ is the eigenvalue matrix of \mathbf{R} with the eigenvalues $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_N \geq 0$. The following matrix inversion formula, verified in the Appendix D, namely,

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{B} (\mathbf{D} \mathbf{A}^{-1} \mathbf{B} + \mathbf{C}^{-1})^{-1} \mathbf{D} \mathbf{A}^{-1}, \quad (1.119)$$

readily yields the inverse of the noise covariance in the following form:

$$\begin{aligned} \mathbf{R}^{-1} &= (\mathbf{U} \mathbf{\Phi} \mathbf{U}^H + \sigma^2 \mathbf{I})^{-1} \\ &= \sigma^{-2} \left[\mathbf{I}_N - \mathbf{U} (\mathbf{U}^H \mathbf{U} + \sigma^2 \mathbf{\Phi}^{-1})^{-1} \mathbf{U}^H \right] \\ &= \sigma^{-2} \left[\mathbf{I}_N - \mathbf{U} (\mathbf{I}_N + \sigma^2 \mathbf{\Phi}^{-1})^{-1} \mathbf{U}^H \right] \\ &= \sigma^{-2} [\mathbf{I}_N - \mathbf{U} \mathbf{\Lambda} \mathbf{U}^H], \end{aligned} \quad (1.120)$$

where

$$\mathbf{\Lambda} = \begin{bmatrix} \frac{\lambda_1}{\lambda_1 + \sigma^2} & 0 & \dots & 0 \\ 0 & \frac{\lambda_2}{\lambda_2 + \sigma^2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{\lambda_N}{\lambda_N + \sigma^2} \end{bmatrix}. \quad (1.121)$$

Note that $\lambda_i/(\lambda_i + \sigma^2) \cong 1$ if $\lambda_i \gg \sigma^2$ and $\lambda_i/(\lambda_i + \sigma^2) \cong \lambda_i/\sigma^2$ if $\lambda_i \ll \sigma^2$. From these observations, noted first by Kirsteins and Tufts, \mathbf{R}^{-1} can be approximated by a reduced rank matrix, denoted by \mathbf{R}_k^{-1} , using the principal component method. The reduced rank k of \mathbf{R}^{-1} is selected in accordance with the condition,

$$\lambda_k \geq t\sigma^2 \quad \text{and} \quad \lambda_{k+1} < t\sigma^2,$$

where t is a specified number greater than 1. According to the above selected value of k , the matrices $\mathbf{\Lambda}$ and \mathbf{U} are partitioned as follows:

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{\Lambda}_k & \mathbf{0} \\ \mathbf{0} & \mathbf{\Lambda}_r \end{bmatrix}, \quad \mathbf{U} = [\mathbf{U}_k, \mathbf{U}_r], \quad (1.122)$$

where $\mathbf{\Lambda}_k$ is a $k \times k$ diagonal matrix, $\mathbf{\Lambda}_r$ is a $(N-k) \times (N-k)$ diagonal matrix, \mathbf{U}_k is a $N \times k$ matrix and \mathbf{U}_r is a $N \times (N-k)$ matrix. In terms of the partitions of matrices $\mathbf{\Lambda}$ and \mathbf{U} , the covariance matrix inverse \mathbf{R}^{-1} in (1.120) becomes

$$\mathbf{R}^{-1} = \sigma^{-2} [\mathbf{I}_N - \mathbf{U}_k \mathbf{\Lambda}_k \mathbf{U}_k^H] - \sigma^{-2} \mathbf{U}_r \mathbf{\Lambda}_r \mathbf{U}_r^H, \quad (1.123)$$

where $\mathbf{\Lambda}_k \cong \mathbf{I}_k$ and

$$\mathbf{\Lambda}_r \cong \frac{1}{\sigma^2} \begin{bmatrix} \lambda_{k+1} & 0 & \dots & 0 \\ 0 & \lambda_{k+2} & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & \lambda_N \end{bmatrix}. \quad (1.124)$$

Now, in the principal component method \mathbf{R}^{-1} is approximated by the matrix \mathbf{R}_k^{-1} , given by

$$\begin{aligned} \mathbf{R}_k^{-1} &= \sigma^{-2} [\mathbf{I}_N - \mathbf{U}_k \mathbf{\Lambda}_k \mathbf{U}_k^H] + \mathbf{O}(\sigma^{-4}) \\ &\cong \sigma^{-2} [\mathbf{I}_N - \mathbf{U}_k \mathbf{U}_k^H] \\ &= \sigma^{-2} \mathbf{P}_k, \end{aligned} \quad (1.125)$$

where $\mathbf{O}(\cdot)$ denotes, "order of", $\mathbf{\Lambda}_k \cong \mathbf{I}_k$ and $\mathbf{P}_k = (\mathbf{I}_N - \mathbf{U}_k \mathbf{U}_k^H)$ represents the projection operator of the subspace which is orthogonal to the principal noise component subspace with projection operator $\mathbf{U}_k \mathbf{U}_k^H$.

Appendix A: Householder Transformation

Let $\mathbf{z} = [z_1, z_2, \dots, z_n]^T$ be a complex $MN \times 1$ vector, $\eta = \mathbf{z}^H \mathbf{z}$, $\mathbf{e}_1 = [1, 0, \dots, 0]^T$ and $\mathbf{v} = \mathbf{z} + \eta \mathbf{e}_1 e^{j\theta}$, where $\theta = \tan^{-1} \frac{y_1}{x_1}$ and $z_1 = x_1 + jy_1$. The Housholder transformation matrix \mathbf{H} is given by

$$\mathbf{H} = \mathbf{I}_n - \frac{2\mathbf{v}\mathbf{v}^H}{\mathbf{v}^H \mathbf{v}}. \quad (1.126)$$

Then $\mathbf{H}^H \mathbf{H} = \mathbf{I}_n$ and $\mathbf{H} \mathbf{z} = -\eta \mathbf{e}_1 e^{j\theta}$.

Proof

The first equality is immediate. Next

$$\begin{aligned}
\mathbf{H} \mathbf{z} &= \mathbf{z} - \frac{2\mathbf{v}\mathbf{v}^H \mathbf{z}}{\mathbf{v}^H \mathbf{v}} = \mathbf{z} - (\mathbf{z} + \eta \mathbf{e}_1 e^{j\theta}) \frac{2(\mathbf{z} + \eta \mathbf{e}_1 e^{j\theta})^H \mathbf{z}}{(\mathbf{z} + \eta \mathbf{e}_1 e^{j\theta})^H (\mathbf{z} + \eta \mathbf{e}_1 e^{j\theta})} \\
&= \mathbf{z} - (\mathbf{z} + \eta \mathbf{e}_1 e^{j\theta}) \frac{2(\eta^2 + \eta \mathbf{e}_1^H \mathbf{z} e^{-j\theta})}{\eta^2 + \eta e^{-j\theta} \mathbf{e}_1^H \mathbf{z} + \eta e^{+j\theta} \mathbf{z}^H \mathbf{e}_1 + \eta^2} \\
&= \mathbf{z} - (\mathbf{z} + \eta \mathbf{e}_1 e^{j\theta}) \frac{(\eta + \mathbf{e}_1^H \mathbf{z} e^{-j\theta})}{(\eta + \text{Re}\{\mathbf{e}_1^H \mathbf{z} e^{-j\theta}\})} \\
&= \mathbf{z} - (\mathbf{z} + \eta \mathbf{e}_1 e^{j\theta}) \frac{\eta + (x_1 \cos \theta + y_1 \sin \theta) + j(y_1 \cos \theta - x_1 \sin \theta)}{\eta + (x_1 \cos \theta + y_1 \sin \theta)} \\
&= \eta \mathbf{e}_1 e^{j(\theta+\pi)},
\end{aligned} \tag{1.127}$$

since $y_1 \cos \theta - x_1 \sin \theta = \frac{x_1 y_1}{\sqrt{x_1^2 + y_1^2}} - \frac{x_1 y_1}{\sqrt{x_1^2 + y_1^2}} = 0$. QED

Appendix B: Derivation of the pdf of η^{-1}

The pdf of $\eta^{-1} = \mathbf{z}_1^H \mathbf{z}_1 - \mathbf{z}_1^H \mathbf{Z}(\mathbf{Z}^H \mathbf{Z})^{-1} \mathbf{Z}^H \mathbf{z}_1$ is derived in this appendix by a use of the Cochran's theorem [37].

Cochran's Theorem: Assume that \mathbf{x} is a $n \times 1$ zero mean, white Gaussian, random vector, i.e. $N_n(\mathbf{0}, \mathbf{I}_n)$ and \mathbf{C} is a symmetric matrix. Then

$$\mathbf{x}^H \mathbf{C} \mathbf{x} \stackrel{d}{=} \chi_k^2$$

if and only if

$$\mathbf{C}^2 = \mathbf{C} \quad \text{and} \quad \text{rank}(\mathbf{C}) = k.$$

Next, η^{-1} can be arranged to be in the quadratic form $\mathbf{x}^H \mathbf{C} \mathbf{x}$ as follows:

$$\begin{aligned}
\eta^{-1} &= \mathbf{z}_1^H \mathbf{z}_1 - \mathbf{z}_1^H \mathbf{Z}(\mathbf{Z}^H \mathbf{Z})^{-1} \mathbf{Z}^H \mathbf{z}_1 \\
&= \mathbf{z}_1^H (\mathbf{I} - \mathbf{Z}(\mathbf{Z}^H \mathbf{Z})^{-1} \mathbf{Z}^H) \mathbf{z}_1 \\
&= \mathbf{z}_1^H (\mathbf{I} - \mathbf{P}) \mathbf{z}_1,
\end{aligned} \tag{1.128}$$

where

$$\mathbf{P} = \mathbf{Z}(\mathbf{Z}^H \mathbf{Z})^{-1} \mathbf{Z}^H \tag{1.129}$$

is evidently a projection matrix with rank equal to $N-1$. Note also that $(\mathbf{I} - \mathbf{P})$ is also a projection matrix of rank $(K - N + 1)$. Since the entries of \mathbf{z}_1 are i.i.d. $N(0,1)$ and $(\mathbf{I} - \mathbf{P})^2 = (\mathbf{I} - \mathbf{P})$, one has by Cochran's theorem that

$$\mathbf{z}_1^H (\mathbf{I} - \mathbf{P}) \mathbf{z}_1 \stackrel{d}{=} \chi_{K-N+1}^2. \tag{1.130}$$

Appendix C: Derivation of the pdf of ζ

The pdf of ζ is derived in this appendix. One can also find a similar derivation in [5]. By (1.84)

$$\zeta = \left(1 + \mathbf{z}_1^H \mathbf{Z} (\mathbf{Z}^H \mathbf{Z})^{-2} \mathbf{Z}^H \mathbf{z}_1\right)^{-1} . \quad (1.131)$$

Next let

$$\delta = \mathbf{z}_1^H \mathbf{Z} (\mathbf{Z}^H \mathbf{Z})^{-2} \mathbf{Z}^H \mathbf{z}_1 . \quad (1.132)$$

Since \mathbf{Z} is positive definite, one can find the QR decomposition of \mathbf{Z} by the Householder transformation such that

$$\mathbf{Z} = \mathbf{Q}_Z \mathbf{R}_Z , \quad (1.133)$$

where \mathbf{Q}_Z is a $K \times (N-1)$ matrix with $\mathbf{Q}_Z^H \mathbf{Q}_Z = \mathbf{I}_{N-1}$ and \mathbf{R}_Z is a $(N-1) \times (N-1)$ lower triangular matrix. The QR decomposition is performed by successively applying the Householder transformation to each row of the matrix \mathbf{Q}_Z . From the top row to the bottom row, one row at a time, one transforms the matrix \mathbf{Z} . After the $(N-1)$ -st transformation, one obtains

$$\mathbf{Z} \mathbf{Q}_1 \mathbf{Q}_2 \dots \mathbf{Q}_{N-1} = \mathbf{R}_Z = \begin{bmatrix} \|\mathbf{z}_K\| e^{j\theta_1} & 0 & \dots & 0 \\ * & \|\mathbf{z}_{K-1}\| e^{j\theta_2} & 0 & \dots & 0 \\ \vdots & & & & \vdots \\ * & \dots & \dots & \dots & \|\mathbf{z}_{K-N+2}\| e^{j\theta_{K-N+2}} \end{bmatrix} , \quad (1.134)$$

where \mathbf{Q}_i is the Householder transformation matrix with respect to the i^{th} row of length $K-i+1$ and $\|\mathbf{z}_j\|$ represents the vector norm of length j .

By a substitution of $\mathbf{Z} = \mathbf{Q}_Z \mathbf{R}_Z$ into (1.132), δ becomes

$$\begin{aligned} \delta &= \mathbf{z}_1^H \mathbf{Q}_Z \mathbf{R}_Z (\mathbf{R}_Z^H \mathbf{R}_Z)^{-2} \mathbf{R}_Z^H \mathbf{Q}_Z^H \mathbf{z}_1 \\ &= \tilde{\mathbf{z}}_1^H \mathbf{R}_Z^{-H} \mathbf{R}_Z^{-1} \tilde{\mathbf{z}}_1 , \end{aligned} \quad (1.135)$$

where $\tilde{\mathbf{z}}_1 = \mathbf{Q}_Z^H \mathbf{z}_1$. Next, apply the Householder transform by picking a unitary matrix \mathbf{Q}_q such that

$$\mathbf{Q}_q \tilde{\mathbf{z}}_1 = \sqrt{\tilde{\mathbf{z}}_1^H \tilde{\mathbf{z}}_1} \mathbf{e}_{N-1} e^{j \arg[\tilde{\mathbf{z}}_1]_{N-1}} . \quad (1.136)$$

Then, one obtains the following result:

$$\begin{aligned} \delta &\underset{d}{=} \tilde{\mathbf{z}}_1^H \mathbf{Q}_q \mathbf{R}_Z^{-H} \mathbf{R}_Z^{-1} \mathbf{Q}_q \tilde{\mathbf{z}}_1 \\ &= \tilde{\mathbf{z}}_1^H \tilde{\mathbf{z}}_1 \mathbf{e}_{N-1}^H (\mathbf{R}_Z^{-H} \mathbf{R}_Z^{-1}) \mathbf{z}_1 \\ &= \tilde{\mathbf{z}}_1^H \tilde{\mathbf{z}}_1 \left[(\mathbf{R}_Z^{-H} \mathbf{R}_Z^{-1}) \right]_{N-1, N-1} \\ &\underset{d}{=} \frac{\chi_{N-1}^2}{\chi_{K-N+2}^2} . \end{aligned} \quad (1.137)$$

Finally by (1.134) one finds that $\mathbf{R}_Z^{-H} \mathbf{R}_Z^{-1}$ is given by

$$\mathbf{R}_Z^{-H} \mathbf{R}_Z^{-1} = \begin{bmatrix} \frac{1}{\|\mathbf{z}_K\|^2} & 0 & \dots & 0 \\ * & \frac{1}{\|\mathbf{z}_{K-1}\|^2} & 0 & \dots & 0 \\ \vdots & & & & \vdots \\ * & \dots & \dots & \frac{1}{\|\mathbf{z}_{K-N+2}\|^2} \end{bmatrix} \stackrel{d}{=} \begin{bmatrix} \frac{1}{\chi_K^2} & 0 & \dots & 0 \\ * & \frac{1}{\chi_{K-1}^2} & 0 & \dots & 0 \\ \vdots & & & & \vdots \\ * & \dots & \dots & \frac{1}{\chi_{K-N+2}^2} \end{bmatrix} \quad (1.138)$$

Then by (1.131), (1.137) and the fact that if $Y \stackrel{d}{=} \chi_n^2$ and $Z \stackrel{d}{=} \chi_m^2$ are independent, then one obtains $Y/(Y+Z) \stackrel{d}{=} \beta(\frac{1}{2}n, \frac{1}{2}m)$, where $\beta(a, b)$ is a beta distributed variate. Hence the final result is given by

$$\zeta = \frac{1}{1 + \delta_d} = \beta(K - N + 2, N - 1) \quad . \quad (1.139)$$

Appendix D: Verification of the Inversion Formula in (1.119)

$$\begin{aligned} & \left[\mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B} \left(\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1} \right)^{-1} \mathbf{D}\mathbf{A}^{-1} \right] (\mathbf{A} + \mathbf{B}\mathbf{C}\mathbf{D}) \\ &= \mathbf{I} - \mathbf{A}^{-1}\mathbf{B} \left(\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1} \right)^{-1} \mathbf{D} - \mathbf{A}^{-1}\mathbf{B} \left(\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1} \right)^{-1} \mathbf{D}\mathbf{A}^{-1}\mathbf{B}\mathbf{C}\mathbf{D} + \mathbf{A}^{-1}\mathbf{B}\mathbf{C}\mathbf{D} \\ &= \mathbf{I} - \mathbf{A}^{-1}\mathbf{B} \left(\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1} \right)^{-1} \mathbf{C}^{-1}\mathbf{C}\mathbf{D} + \mathbf{A}^{-1}\mathbf{B} \left(\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1} \right)^{-1} \mathbf{D}\mathbf{A}^{-1}\mathbf{B}\mathbf{C}\mathbf{D} + \mathbf{A}^{-1}\mathbf{B}\mathbf{C}\mathbf{D} \\ &= \mathbf{I} - \mathbf{A}^{-1}\mathbf{B} \left[\left(\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1} \right)^{-1} \mathbf{C}^{-1} + \left(\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1} \right)^{-1} \mathbf{D}\mathbf{A}^{-1}\mathbf{B} - \mathbf{I} \right] \mathbf{C}\mathbf{D} \\ &= \mathbf{I} - \mathbf{A}^{-1}\mathbf{B} \left[\left(\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1} \right)^{-1} \left(\mathbf{D}\mathbf{A}^{-1}\mathbf{B} + \mathbf{C}^{-1} \right) - \mathbf{I} \right] \mathbf{C}\mathbf{D} = \mathbf{I} \quad \text{QED} \end{aligned}$$

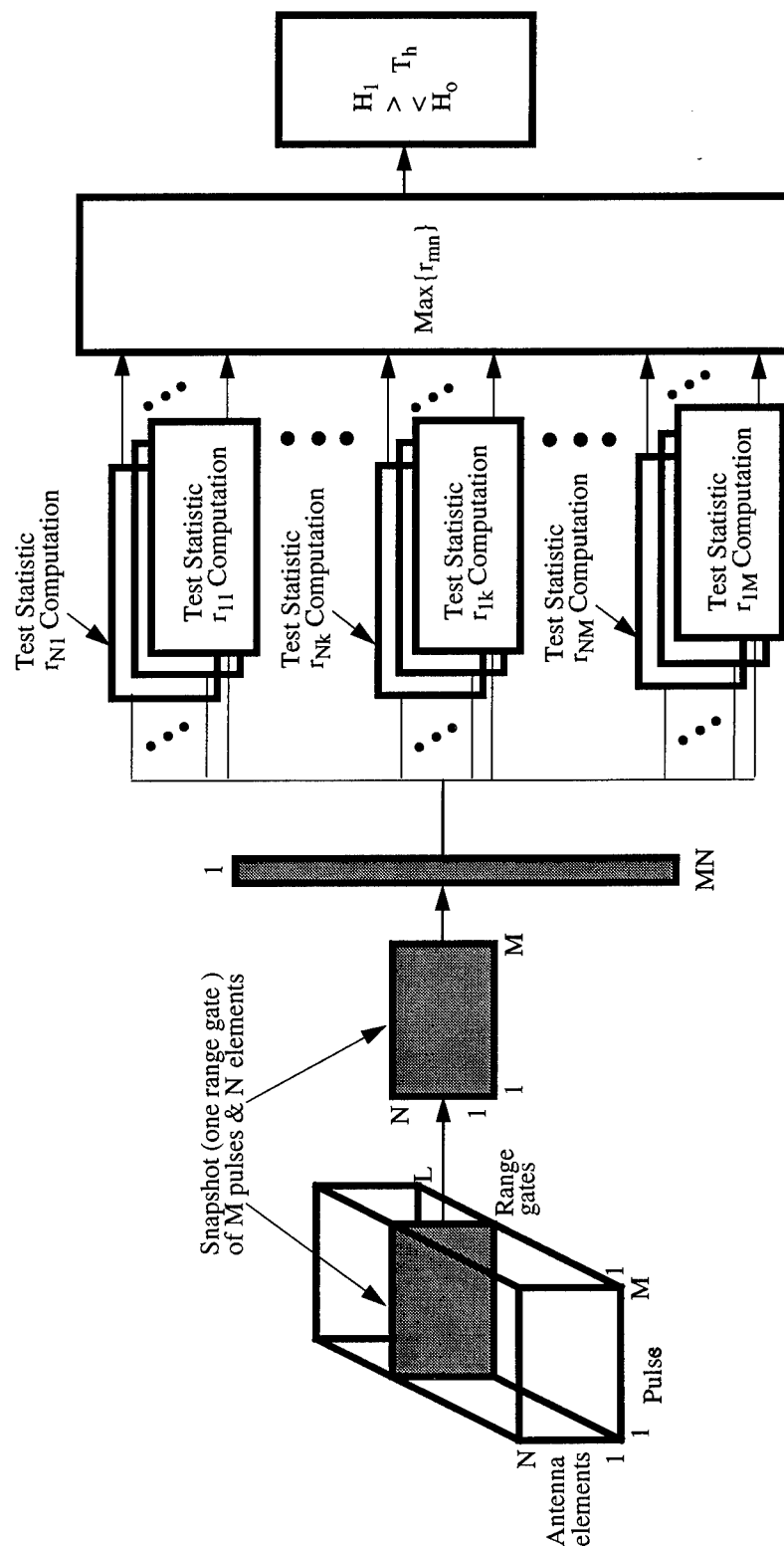


Figure 1 System Block Diagram of Fully Adaptive STAP Estimator-Detector

Figure 1.10: Fully adaptive STAP.

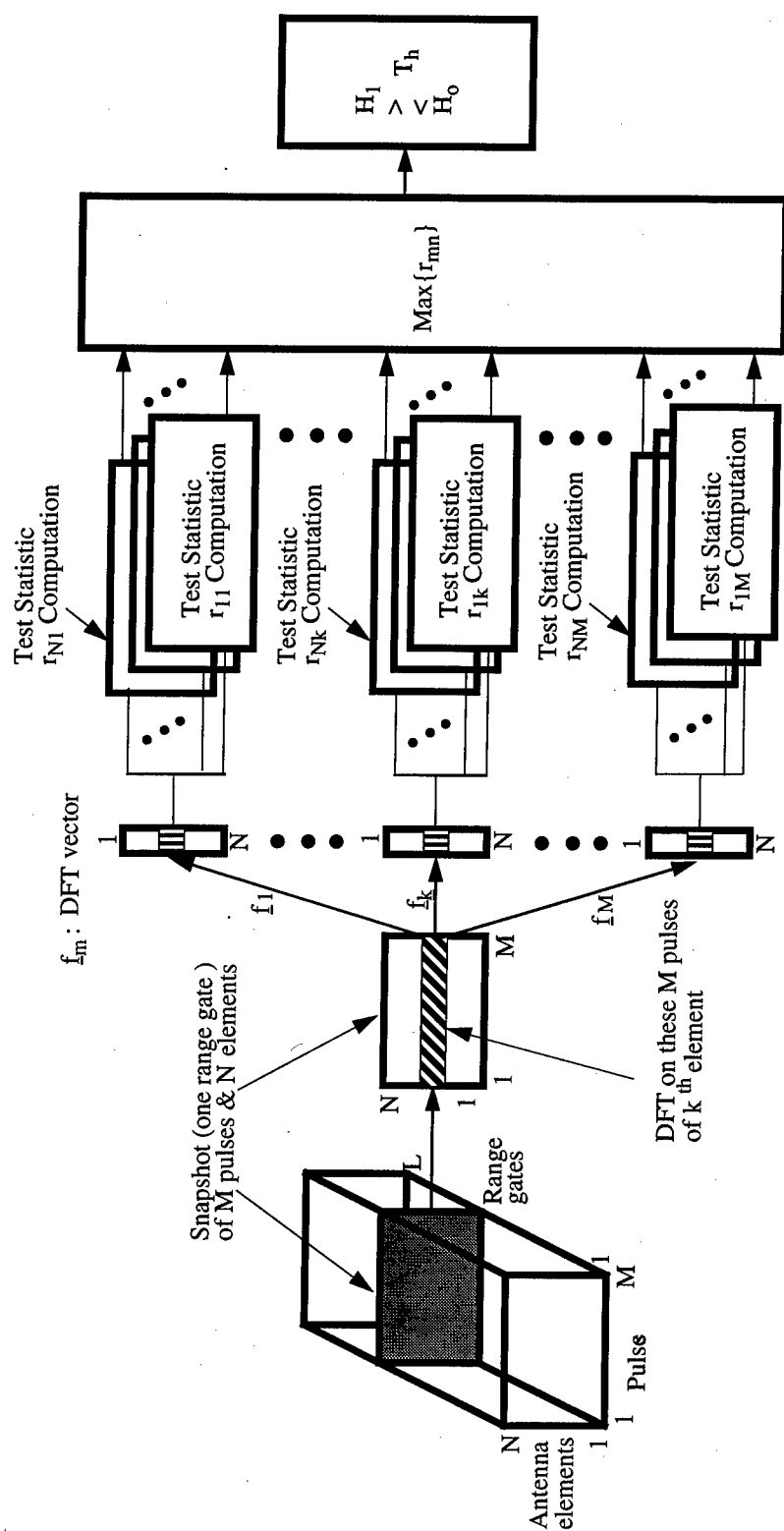


Figure 2 System Block Diagram of Element - Space Post-Doppler STAP Estimator-Detector

Figure 1.11: Element-space, post-doppler STAP.

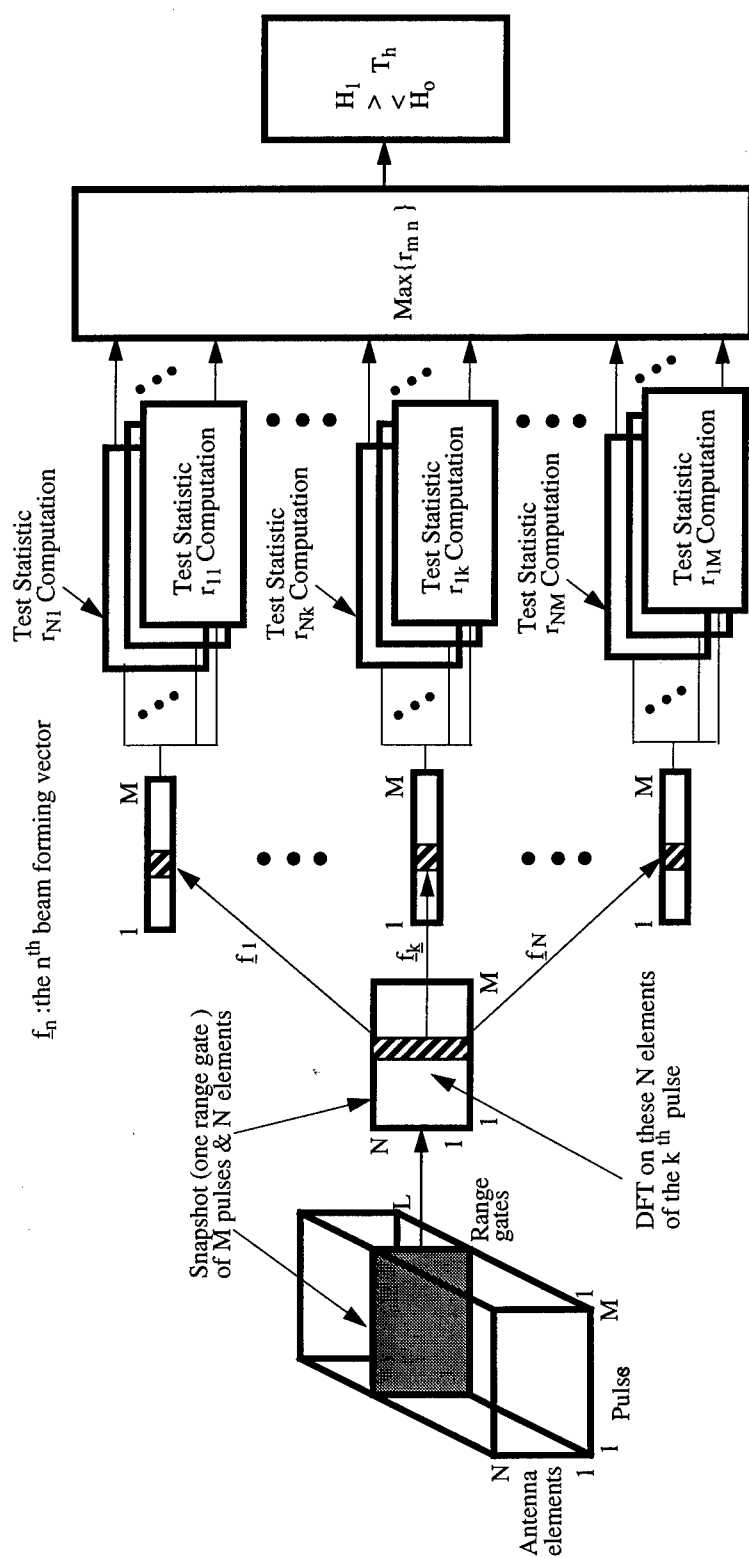


Figure 3 System Block Diagram of Beam-Space Post-Doppler Partial STAP Estimator-Detector

Figure 1.12: Beam-space, post-doppler partial STAP.

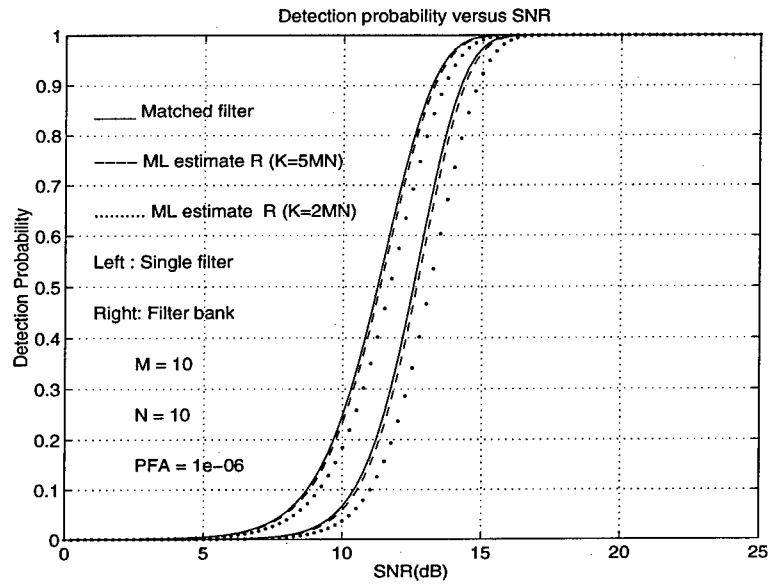


Figure 1.13: Probability of detection of fully adaptive STAP ($M=10, N=10$).

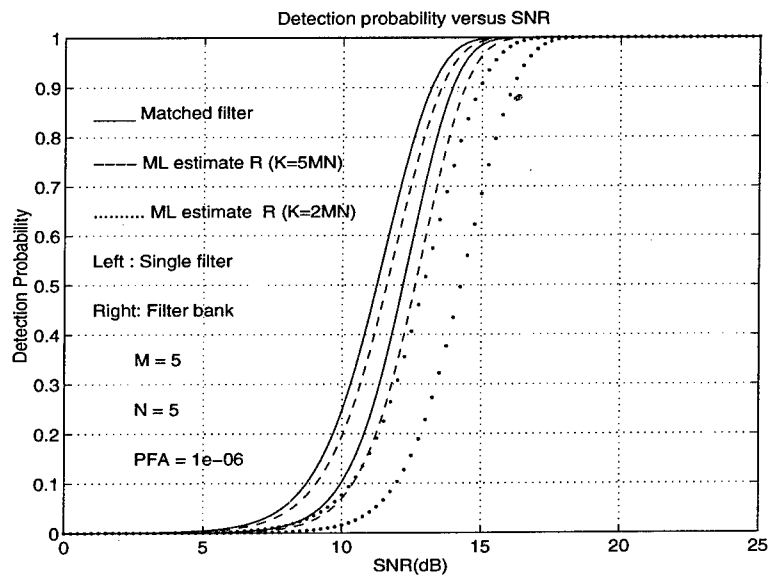


Figure 1.14: Probability of detection of fully adaptive STAP ($M=5, N=5$).

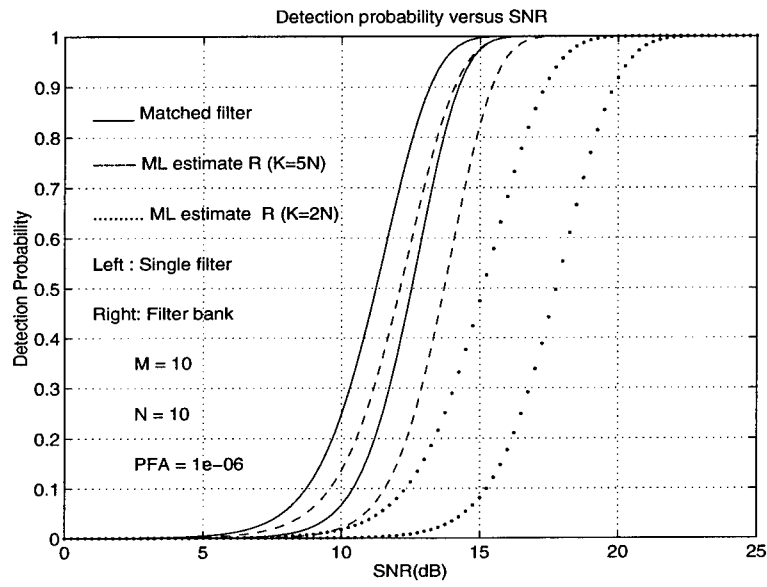


Figure 1.15: Probability of detection of partially adaptive STAP ($M=10, N=10$).

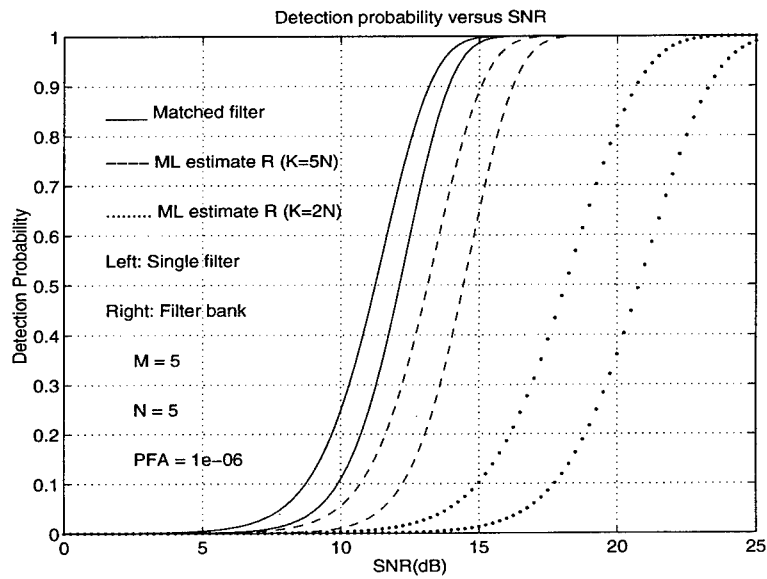


Figure 1.16: Probability of detection of partially adaptive STAP ($M=5, N=5$).

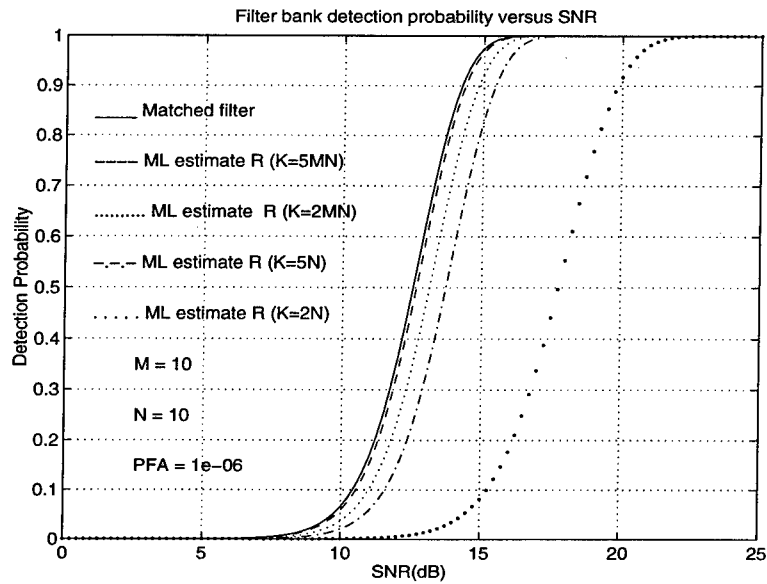


Figure 1.17: Comparison of the probability of detection between fully and partially adaptive STAP ($M=10, N=10$)

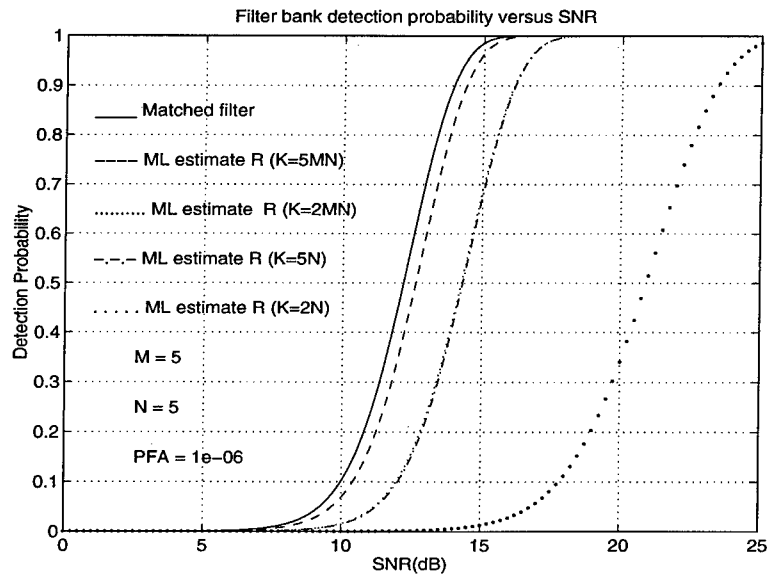


Figure 1.18: Comparison of the probability of detection between fully and partially adaptive STAP ($M=5, N=5$)

Bibliography

- [1] S.P. Applebaum, "Adaptive arrays," Technical Report SPL TR 66-1, Syracuse University Research Corporation, August 1966.
- [2] L.W. Brooks and I.S. Reed, "Equivalence of the likelihood ratio processor, the maximum signal-to-noise ratio filter and the wiener filter," *IEEE Trans. Aerosp. Electron. Syst.*, AES-8:690-692, September 1972.
- [3] L.L. Scharf, *Statistical Signal Processing*. Addison-Wesley, Reading, MA, 1991.
- [4] H.L. Van Trees, *Detection, Estimation, and Modulation Theory*, volume 1, John Wiley and Sons, New York, NY, 1968.
- [5] S. Haykin and A. Steinhardt, editors, *Adaptive Radar Detection and Estimation*. John Wiley and Sons, New York, NY, 1992.
- [6] J.S. Goldstein, "Performance measures for optimal constrained beamformers," *IEEE Trans. Antenn. Propagat.*, Submitted 1995.
- [7] S.P. Applebaum and D.J. Chapman, "Adaptive arrays with main beam constraints," *IEEE Trans. Antenn. Propagat.*, AP-24(5):650-662, September 1976.
- [8] L.J. Griffiths and C.W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans. Antenn. Propagat.*, AP-30(1):27-34, January 1982.
- [9] J.S. Goldstein and I.S. Reed, "Subspace selection for partially adaptive sensor array processing," *IEEE Trans. on Aerosp. Electron. Syst.*, Accepted for publication, 1996.
- [10] J.S. Goldstein and I.S. Reed, "Reduced rank Wiener filtering: Theory and applications," *IEEE Trans. Signal Processing*, Submitted 1995.
- [11] J.S. Goldstein, P.A. Zulch, and I.S. Reed, "Reduced rank space-time adaptive radar processing," In *Proc. IEEE ICASSP*, Atlanta, GA, May to appear, 1996.
- [12] J.S. Goldstein, S.M. Kogon, I.S. Reed, D.B. Williams, and E.J. Holder, "Partially adaptive radar signal processing: The cross-spectral approach," In *Proc. 29th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, November 1995.
- [13] I.P. Kirsteins and D.W. Tufts, "Adaptive detection using a low rank approximation to a data matrix," *IEEE Trans. Aerosp. Electron. Syst.*, 30(1):55-67, January 1994.

- [14] W. Harrison and D. Tufts, "Rapidly adaptive mainbeam jammer nulling," In *Proc. ASAP Workshop*, volume ASAP-3 Addendum. MIT Lincoln Laboratory, July 1994.
- [15] A. Haimovich, M. Berlin, and X. Wu, "Performance of the eigencanceler: Eigenanalysis space-time adaptive radar," In *Proc. ASAP Workshop*, volume ASAP-3, pages 419-445. MIT Lincoln Laboratory, April 1995.
- [16] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, 1:211-218, 1936.
- [17] L.E. Brennan and I.S. Reed, "Theory of adaptive radar," *IEEE Trans. Aerosp. Electron. Syst.*, 9:237-251, March 1973.
- [18] J. Ward, "Space-time adaptive processing for airborne radar," Technical Report 1015, MIT Lincoln Laboratory, December 1994.
- [19] W.F. Gabriel, "Using spectral estimation techniques in adaptive processing antenna systems," *IEEE Trans. Antenn. Propagat.*, AP-34(3):291-300, March 1986.
- [20] B.D. Van Veen, "Eigenstructure based partially adaptive array design," *IEEE Trans. Antenn. Propagat.*, 36(3):357-362, March 1988.
- [21] D.O. Carhoun, R.A. Games, and R.T. Williams, "A principal components sidelobe cancellation algorithm," Technical Report M90-82, MITRE, Dept. D082, November 1990.
- [22] K.A. Byerly and R.A. Roberts, "Output power based partially adaptive array design," In *Proc. 23rd Asilomar Conf. Signals, Syst. Comput.*, pages 576-580, Pacific Grove, CA, November 1989.
- [23] G.W. Titi, "An overview of the ARPA/Navy mountaintop program," In *Proc. of the IEEE Adapt. Ant. Systems Symp.*, Long Island, NY, November 1994.
- [24] E. J. Kelly, "An Adaptive detection algorithm," *IEEE Trans. on Aerospace and Electronic Systems*, AES-22, March 1986.
- [25] I. S. Reed, J. D. Mallett, and L. E. Brennan, "Rapid Convergence Rate in Adaptive Arrays," *IEEE Trans. on Aerospace and Electronic Systems*, AES-10, Nov. 1974.
- [26] Wai-Sheon Chen and I. S. Reed, "A new CFAR Detection Test for Radar", *Digital Signal Processing*, Vol. No. 4, Academic Press Inc., San Diego, CA, Oct., 1991.
- [27] F. Robey, D. Fuhrmann, E. Kelly and R. Nitzberg, "A CFAR Adaptive matched filter detector", *IEEE Trans. on Aerospace and Electronic Systems*, Jan. 1992.
- [28] H. Hotelling, "Analysis of a Complex of Statistical Variables into Principal Components," *J. Educ. Psychol.*, 24, 1933.
- [29] R. J. Muirhead, *Aspects of Multivariate Statistical Theory* J. Wiley and Sons, Inc. New York, 1982.
- [30] Eric K. L. Hung and Ross M. Turner, "A Fast Beamforming Algorithm for Large Arrays" *IEEE Trans. on Aerospace and Electronic Systems*, July, 1983.

- [31] Xiaoli Yu and Irving S. Reed, "Adaptive Detection of Signals with Linear Feature Mappings and Representations", to be published in *IEEE Trans. on Signal Processing*, 1995.
- [32] I. S. Reed and Y. L. Gau, "A Simultaneous CFAR Detection and ML Estimation STAP Algorithm for Airborne Radar", submitted for publication in 1995.
- [33] J. Ward, "On Brennan's rule for noninteger interpulse motions", M.I.T. Lincoln Laboratory memo MTN-74, 1 February 1994.
- [34] Allan Steinhardt, "Householder Transforms in Signal Processing", *IEEE ASSP Magazine*, July, 1988.
- [35] Gilbert Strang, *Linear Algebra and its Applications*, Harcourt Brace Jovanovich, Inc. 1988.
- [36] S. Haykin, J. Litva and T. Shepherd, *Radar Array Processing*, Springer-Verlag, 1993.
- [37] Fang Kai-Tai Zhang Yao-Ting, *Generalized Multivariate Analysis* Springer-Verlag Science Press, 1990.
- [38] Don H. Johnson, Dan E. Dudgeon, *Array Signal Processing*, Prentice Hall Inc., New Jersey, 1993.

Part 2

Robust Detection and Estimation with Alpha-Stable Distributions for STAP Applications

Principal Investigator: Professor Chrysostomos L. Nikias

**Collaborators: Russell Lambert, Xinyu (Jack) Ma, Dr. Panagiotis Tsakalides,
and Professor George A. Tsihrintzis**

Spikes due to clutter sources such as mountains, forests or ocean waves, and glints due to reflections from large flat surfaces such as buildings or vehicles are usually present in radar returns. Their presence obscures the target detection capability of a radar system and degrades the system's performance. Preprocessing to suppress the spikes/glints in the radar returns and minimize their effect on the radar target detection performance can be efficiently done only on the basis of appropriate statistical modeling.

Very recently, a statistical model of impulsive interference has been proposed, which is based on the theory of symmetric alpha-stable ($S\alpha S$) random processes [1]. The model is of a statistical-physical nature and has been shown to arise under very general assumptions and to describe a broad class of impulsive interference. In particular, it has been shown in [1] that the first order distribution of the amplitude of the radar return follows a $S\alpha S$ law, while the first-order joint distribution of the quadrature components of the envelope of the radar return follows an isotropic stable law. This model is mathematically more appealing than existing models for impulsive interference.

In this part, we address the solution of the target detection and signal parameter estimation problems through the use of radar array sensor data retrieved in the presence of impulsive interference (noise, clutter, or jamming.) First, we describe new methods on the modeling of the amplitude statistics of airborne radar clutter by means of alpha-stable distributions. Then, we introduce target detection and joint angle and Doppler estimation techniques from radar measurements retrieved in the presence of impulsive noise modeled as a multivariate sub-Gaussian random process. The results are of great importance in the study of space-time adaptive processing (STAP) for airborne pulse Doppler radar arrays operating in impulsive interference environments.

This part includes the results in the area of alpha-stable modeling and is organized as follows:

1. Introduction of the statistical model, based on the class of symmetric α -stable ($S\alpha S$) distributions [Section 2.1];
2. Characterization and development of methods for parameter estimation of $S\alpha S$ distributions by means of order statistics, fractional lower-order and negative-order moments. Modeling of the amplitude statistics of radar clutter by means of $S\alpha S$ distributions and estimation of the parameters of the stable distributions from real clutter of the Mountain Top Database [Section 2.2];
3. Development of an adaptive matched-filter detector for the case of alpha-stable, sub-Gaussian noise. Comparison of the performance of the proposed detector to the that of the optimum detector under completely known signal and Gaussian noise characteristics [Section 2.3];
4. Development of the Cauchy Beamformer for joint spatial- and Doppler-frequency maximum likelihood estimation. The Cauchy Beamformer is shown to be very robust in a wide range of additive impulsive noise environments. [Section 2.4.1];
5. Development of a new joint spatial- and doppler-frequency high-resolution estimation technique based on the eigendecomposition of the *covariation matrix* of the space-time radar measurements [Section 2.4.2].

2.1 Alpha-Stable Random Variables and Processes

Gaussian distributions and processes have long been accepted as useful tools for stochastic modeling. In this section, we introduce a statistical model based on the class of *symmetric α -stable ($S\alpha S$) distributions* which is well-suited for describing signals that are impulsive in nature. A review of the state of the art on stable processes from a statistical point of view is provided by a collection of papers edited by Cambanis, Samorodnitsky and Taqqu [2]. Several statisticians including Cambanis, Zolotarev, Weron, *et al.* have published extensively on the theory and applications of stable processes. They studied the properties of stable processes [3, 4, 5], their spectral representation [6, 7], as well as prediction and linear filtering problems [8, 9]. Textbooks in the area were written by Samorodnitsky and Taqqu [10] and by Janicki and Weron [11]. An extensive review of stable processes from a signal processing point of view can be found in a tutorial paper by Shao and Nikias [12] as well as in a monogram written by the same authors [1].

2.1.1 The Class of Real $S\alpha S$ Distributions

The symmetric α -stable ($S\alpha S$) distribution is best defined by its characteristic function

$$\varphi(\omega) = \exp(j\delta\omega - \gamma|\omega|^\alpha), \quad (2.1)$$

where α is the *characteristic exponent* restricted to the values $0 < \alpha \leq 2$, δ ($-\infty < \delta < \infty$) is the *location parameter*, and γ ($\gamma > 0$) is the *dispersion* of the distribution. For values of α in the interval $(1, 2]$, the location parameter δ corresponds to the mean of the $S\alpha S$ distribution, while for $0 < \alpha \leq 1$, δ corresponds to its median. The dispersion parameter γ determines the spread of the

distribution around its location parameter δ , similar to the variance of the Gaussian distribution. The characteristic exponent α is the most important parameter of the $S\alpha S$ distribution and it determines the shape of the distribution.

A stable distribution is called *standard* if $\delta = 0$ and $\gamma = 1$. Clearly, if a random variable X is stable with parameters α, γ, δ , then $(X - \delta)/\gamma^{1/\alpha}$ is standard with characteristic exponent α . The standard $S\alpha S$ density functions for a few values of the characteristic exponent α are shown in Figure 2.1.

By letting α take the values 1 and 2, we get two important special cases of $S\alpha S$ distributions, namely, the *Cauchy* ($\alpha = 1$), and the *Gaussian* ($\alpha = 2$):

Cauchy

$$f_1(\gamma, \delta; x) = \frac{1}{\pi} \frac{\gamma}{\gamma^2 + (x - \delta)^2} \quad (2.2)$$

Gaussian

$$f_2(\gamma, \delta; x) = \frac{1}{\sqrt{4\pi\gamma}} \exp\left[-\frac{(x - \delta)^2}{4\gamma}\right]. \quad (2.3)$$

Unfortunately, no closed form expressions exist for general $S\alpha S$ distributions other than the Cauchy and the Gaussian. However, power series expansions can be derived for $f_\alpha(\gamma, \delta; x)$ [12]. Although the $S\alpha S$ density behaves approximately like a Gaussian density near the origin, its tails decay at a lower rate than the Gaussian density tails. While the Gaussian density has exponential tails, the stable densities have algebraic tails (cf. Figure 2.2). *The smaller the characteristic exponent α is, the heavier the tails of the $S\alpha S$ density.* This implies that random variables following $S\alpha S$ distributions with small characteristic exponents are *highly impulsive*. It is this heavy-tail characteristic that makes the $S\alpha S$ densities appropriate for modeling signals and noise or interference which are impulsive in nature.

$S\alpha S$ densities obey two important properties which further justify their role in data modeling:

- The *stability property*, which states that the random variables X_1, \dots, X_n are independent and symmetrically stable with the same characteristic exponent α if and only if for any constants a_1, \dots, a_n , the linear combination $\sum_{i=1}^n a_i X_i$ is also $S\alpha S$;
- The *generalized central limit theorem*, which states that the family of stable distributions contains all limiting distributions of sums of i.i.d. random variables.

An important difference between the Gaussian and the other distributions of the $S\alpha S$ family is that only moments of order less than α exist for the non-Gaussian $S\alpha S$ family members. The *fractional lower order moments* (FLOM's) of a $S\alpha S$ random variable with zero location parameter and dispersion γ are given by:

$$E|X|^p = C(p, \alpha) \gamma^{\frac{p}{\alpha}} \quad \text{for } 0 < p < \alpha \quad (2.4)$$

where

$$C(p, \alpha) = \frac{2^{p+1} \Gamma(\frac{p+1}{2}) \Gamma(-\frac{p}{\alpha})}{\alpha \sqrt{\pi} \Gamma(-\frac{p}{2})} \quad (2.5)$$

and $\Gamma(\cdot)$ is the Gamma function.

2.1.2 Symmetric Sub-Gaussian Alpha-Stable Processes

A collection of random variables $\{X(t), t \in T\}$ where T is an arbitrary index set, is said to be a $S\alpha S$ stochastic process if for all combinations of distinct indices $t_1, \dots, t_n \in T$, the random variables $X(t_1), \dots, X(t_n)$ are jointly $S\alpha S$ with the same characteristic exponent α . The stochastic process $\{X(t), t \in T\}$ is *stationary* if the random vectors $(X(t_1), \dots, X(t_n))$ and $(X(t_1+s), \dots, X(t_n+s))$ are identically distributed for each choice of $s, t_1, \dots, t_n \in T$. The family of stable processes has many members with mutually exclusive properties. In the following, we present the important class of *sub-Gaussian stable processes* which we will use extensively.

Sub-Gaussian Processes

A stable process $\{X(t), t \in T\}$ is said to be an α -sub-Gaussian process (α -SG(\underline{R})) if for distinct indices $t_1, \dots, t_n \in T$, the random vector $\underline{X} = [X(t_1), \dots, X(t_n)]^T$ has characteristic function of the general form

$$\phi(\underline{\omega}) = \exp\left[-\frac{1}{2}(\underline{\omega}^T \underline{R} \underline{\omega})^{\alpha/2}\right], \quad (2.6)$$

where \underline{R} is a positive-definite matrix, called the *underlying matrix* of the process, $\underline{\omega} = [\omega_1, \dots, \omega_n]^T$, and α takes values in $(1, 2]$. When $\alpha = 2$, \underline{X} is a Gaussian vector with zero mean and covariance matrix \underline{R} . Unfortunately, closed-form expressions for the joint pdf of sub-Gaussian random vectors are known only for the Gaussian ($\alpha = 2$) and Cauchy ($\alpha = 1$) cases:

$$f_G(\underline{X}) = \frac{1}{\sqrt{(2\pi)^L ||\underline{R}||}} \exp(-\underline{X}^T \underline{R}^{-1} \underline{X}) \quad (\text{Gaussian}) \quad (2.7)$$

$$f_C(\underline{X}) = \frac{c ||\underline{R}||^{-1/2}}{[1 + \underline{X}^T \underline{R}^{-1} \underline{X}]^{(L+1)/2}} \quad (\text{Cauchy}), \quad (2.8)$$

where L is the length of the random vector, $||\underline{R}||$ is the determinant of \underline{R} , and $c = \frac{1}{\pi^{(L+1)/2}} \Gamma(\frac{L+1}{2})$.

Sub-Gaussian processes share many common features with Gaussian processes. In fact, sub-Gaussian processes are variance mixtures of Gaussian processes [13]. Specifically, any sub-Gaussian random vector can be expressed in the form

$$\underline{X} = w^{\frac{1}{2}} \underline{G}, \quad (2.9)$$

where w is a positive $\frac{\alpha}{2}$ -stable random variable [14] and \underline{G} is a Gaussian random vector of mean zero and covariance matrix \underline{R} . An important distinction between Gaussian and sub-Gaussian processes is that, while linear spaces of Gaussian random variables may contain non-degenerate independent elements, sub-Gaussian random variables cannot be independent [15].

Sub-Gaussian $S\alpha S$ processes combine the capability to model statistical dependence with the capability to model the presence of outliers in observed time series of various degrees of severity. The example in Figure 2.3 is indicative of the concept. Consider a sub-Gaussian vector of length $L = 100$ and diagonal underlying covariance matrix $\underline{R} = \text{diag}\{1, 1, \dots, 1\}$. Typical realizations of the vector are shown for characteristic exponents $\alpha = 2$ and $\alpha = 1.5$. Clearly, it is difficult to distinguish one vector from the other visually. However, if we look over 1000 independent realizations of the first component of the vector, a clear difference is observed.

In modeling the signals and/or noise for the parameter estimation problem, we need a complex model for the noise samples. We also need to define quantities which describe correlations between

random variables. In the following section, we present the family of isotropic complex $S\alpha S$ distributions and describe their correlation properties by means of the *covariation* quantity.

2.1.3 Complex $S\alpha S$ Random Variables and Covariations

A complex random variable (r.v.) $X = X_1 + jX_2$ is symmetric α -stable ($S\alpha S$) if X_1 and X_2 are jointly $S\alpha S$ and its characteristic function is written as

$$\begin{aligned}\varphi(\omega) &= E \exp[j\Re(\omega X^*)] = E \exp[j(\omega_1 X_1 + \omega_2 X_2)] \\ &= \exp \left[- \int_{S_2} |\omega_1 x_1 + \omega_2 x_2|^\alpha d\Gamma_{X_1, X_2}(x_1, x_2) \right],\end{aligned}\quad (2.10)$$

where $\omega = \omega_1 + j\omega_2$, $\Re[\cdot]$ is the real part operator, and Γ_{X_1, X_2} is a symmetric measure on the unit sphere S_2 , called the *spectral measure* of the random variable X . A complex random variable $X = X_1 + jX_2$ is *isotropic* if and only if (X_1, X_2) has a uniform spectral measure. In this case, the characteristic function of X can be written as

$$\varphi(\omega) = E \exp(j\Re[\omega X^*]) = \exp(-\gamma|\omega|^\alpha), \quad (2.11)$$

where γ ($\gamma > 0$) is the *dispersion* of the distribution.

In the theory of second-order processes, the concept of *covariance* plays an important role in problems of linear prediction, filtering and smoothing. Since $S\alpha S$ processes do not possess finite p th order moments for $p \geq \alpha$, covariances do not exist on the space of $S\alpha S$ random variables. Instead, a quantity called *covariation* plays an analogous role for statistical signal processing problems involving $S\alpha S$ processes to the role played by covariance in the case of second-order processes.

Several complex r.v.'s are jointly $S\alpha S$ if their real and imaginary parts are jointly $S\alpha S$. When $X = X_1 + jX_2$ and $Y = Y_1 + jY_2$ are jointly $S\alpha S$ with $1 < \alpha \leq 2$, the *covariation* of X and Y is defined by

$$[X, Y]_\alpha = \int_{S_4} (x_1 + jx_2)(y_1 + jy_2)^{<\alpha-1>} d\Gamma_{X_1, X_2, Y_1, Y_2}(x_1, x_2, y_1, y_2), \quad (2.12)$$

where we use throughout the convention

$$Y^{<\beta>} = |Y|^{\beta-1} Y^*. \quad (2.13)$$

It can be shown that for every $1 \leq p < \alpha$, the covariation can be expressed as a function of moments [8]

$$[X, Y]_\alpha = \frac{E X Y^{<p-1>}}{E |Y|^p} \gamma_Y, \quad (2.14)$$

where γ_Y is the dispersion of the r.v. Y given by

$$\gamma_Y^{p/\alpha} = \frac{E |Y|^p}{C(p, \alpha)} \quad \text{for } 0 < p < \alpha, \quad (2.15)$$

with

$$C(p, \alpha) = \frac{2^{p+1} \Gamma(\frac{p+2}{2}) \Gamma(-\frac{p}{\alpha})}{\alpha \Gamma(\frac{1}{2}) \Gamma(-\frac{p}{2})}. \quad (2.16)$$

Obviously, from (2.14) it holds that

$$[X, X]_\alpha = \gamma_X. \quad (2.17)$$

Also, the *covariation coefficient* of X and Y is defined by

$$\lambda_{X,Y} = \frac{[X, Y]_\alpha}{[Y, Y]_\alpha}, \quad (2.18)$$

and by using (2.14), it can be expressed as

$$\lambda_{X,Y} = \frac{EXY^{<p-1>}}{E|Y|^p} \quad \text{for } 1 \leq p < \alpha. \quad (2.19)$$

The covariation of complex jointly $S\alpha S$ r.v.'s is not generally symmetric and has the following properties [3]:

P1 If X_1, X_2 and Y are jointly $S\alpha S$, then

$$[aX_1 + bX_2, Y]_\alpha = a[X_1, Y]_\alpha + b[X_2, Y]_\alpha \quad (2.20)$$

for any complex constants a and b .

P2 If Y_1 and Y_2 are independent and X_1, X_2 and Y are jointly $S\alpha S$, then

$$[aX_1, bY_1 + cY_2]_\alpha = ab^{<\alpha-1>}[X_1, Y_1]_\alpha + ac^{<\alpha-1>}[X_1, Y_2]_\alpha \quad (2.21)$$

for any complex constants a, b and c .

P3 If X and Y are independent $S\alpha S$, then $[X, Y]_\alpha = 0$

P4 Let $\{U_i; i = 1, \dots, n\}$ be independent complex $S\alpha S$ r.v.'s with dispersions γ_i . For any complex numbers $\{a_i, b_i; i = 1, \dots, n\}$ form

$$X = a_1U_1 + \dots + a_nU_n,$$

$$Y = b_1U_1 + \dots + b_nU_n.$$

Then

$$\begin{aligned} [X, X]_\alpha &= \gamma_1|a_1|^\alpha + \dots + \gamma_n|a_n|^\alpha, \\ [Y, Y]_\alpha &= \gamma_1|b_1|^\alpha + \dots + \gamma_n|b_n|^\alpha, \\ [X, Y]_\alpha &= \gamma_1a_1b_1^{<\alpha-1>} + \dots + \gamma_na_nb_n^{<\alpha-1>}. \end{aligned} \quad (2.22)$$

2.1.4 Generation of Complex Isotropic $S\alpha S$ Random Variables

The generation of complex isotropic $S\alpha S$ deviates of characteristic exponent α is based on the following proposition found in [10]:

Proposition 2.1 A complex $S\alpha S$ ($\alpha < 2$) random variable $X = X_1 + jX_2$ is isotropic if and only if there exist two i.i.d. zero-mean Gaussian variables G_1 and G_2 and a real stable random variable A of characteristic exponent $\alpha/2$, dispersion $\cos^2(\pi\alpha/4)$ and skewness $\beta = 1$ (we write $A \sim S_{\alpha/2}(\cos^2(\pi\alpha/4), 1)$), independent of (G_1, G_2) such that $(X_1, X_2) \stackrel{d}{=} (A^{1/2}G_1, A^{1/2}G_2)$.

We say that the vector (X_1, X_2) is *sub-Gaussian* with underlying vector (G_1, G_2) . It can be shown that the real and imaginary parts of X are always dependent, unless G_1 and G_2 are degenerate. Hence, every complex isotropic $S\alpha S$ random variable with $\alpha < 2$ can be expressed as

$$X = A^{1/2}(G_1 + jG_2), \quad (2.23)$$

and its generation involves the generation of a real, totally skewed stable random variable. The problem of generating a real stable deviate is studied in [16] and [11]. Here, we present the result for easy reference.

To generate a real standard stable random variable $A \sim S_\alpha(1, \beta)$ of characteristic exponent α , skewness β and unit dispersion $\gamma = 1$, the following representations can be deduced:

$$S(\alpha, \beta, 1) = D_{\alpha, \beta} \frac{\sin \alpha(U - U_0)}{(\cos U)^{1/\alpha}} \left(\frac{\cos(U - \alpha(U - U_0))}{W} \right)^{\frac{1-\alpha}{\alpha}}, \quad \text{for } \alpha \neq 1, \quad (2.24)$$

and

$$S(1, \beta, 1) = \frac{2}{\pi} \left[\left(\frac{\pi}{2} + \beta U \right) \tan U - \beta \ln \left(\frac{\frac{\pi}{2} W \cos U}{\frac{\pi}{2} + \beta U} \right) \right], \quad (2.25)$$

where W is standard exponential with $\Pr\{W > w\} = e^{-w}$, $w > 0$, and U is uniform on $(-\frac{\pi}{2}, \frac{\pi}{2})$. Also, $D_{\alpha, \beta} = [\cos(\arctan(\beta \tan(\pi\alpha/2)))]^{-1/\alpha}$, and $U_0 = -\frac{\pi}{2}\beta[k(\alpha)/\alpha]$ with $k(\alpha) = 1 - |1 - \alpha|$. Then, a stable variate, A_1 , of dispersion γ can be obtained from A by $A_1 = \gamma^{1/\alpha}A$.

2.2 Parameter Estimation with Fractional Lower-Order Moments

The moment theory for $S\alpha S$ processes is based on fractional lower-order statistics. The fractional lower-order moment of a $S\alpha S$ random variable X satisfies:

$$\mathbf{E}(|X|^p) = C_1(p, \alpha) \gamma^{p/\alpha}, \quad \text{for } -1 < p < \alpha, \quad (2.26)$$

where $C_1(p, \alpha) = \frac{2^p \Gamma(\frac{p+1}{2}) \Gamma(1-p/\alpha)}{\sqrt{\pi} \Gamma(1-p/2)}$, α is the characteristic exponent ($0 < \alpha < 2$), γ is the dispersion and $\Gamma(\cdot)$ is the Gamma function. When X is an n dimensional spherically symmetric $S\alpha S$ random variable, $\mathbf{E}(|X|^p)$ is given in [5]:

$$\mathbf{E}(|X|^p) = 2^p \frac{\Gamma(\frac{p+n}{2}) \Gamma(1-\frac{p}{\alpha})}{\Gamma(1-\frac{p}{2}) \Gamma(\frac{n}{2})} \gamma^{p/\alpha}, \quad \text{for } -n < p < \alpha. \quad (2.27)$$

Specifically, when X is an isotropic bivariate $S\alpha S$ random variable, we have:

$$\mathbf{E}(|X|^p) = C_2(p, \alpha) \gamma^{p/\alpha}, \quad \text{for } -2 < p < \alpha, \quad (2.28)$$

where $C_2(p, \alpha) = 2^p \frac{\Gamma(1+\frac{p}{2}) \Gamma(1-\frac{p}{\alpha})}{\Gamma(1-\frac{p}{2})}$. The simplest yet most important class of complex $S\alpha S$ random variables is the class of isotropic bivariate $S\alpha S$.

Notice that the value of p (the order of moment) can be both negative and positive. Using this property, we can estimate the parameters (α, γ) of $S\alpha S$ processes [17].

• **Sinc Function Estimator:**

Letting $0 < p < \min(\alpha, 1)$, such that both positive- and negative-order moments are finite, then:

$$\mathbf{E}(|X|^p)\mathbf{E}(|X|^{-p}) = \frac{2 \tan(p\pi/2)}{\alpha \sin(p\pi/\alpha)}, \quad (2.29)$$

i.e., α can be found by solving the following sinc function equation:

$$\text{sinc}\left(\frac{p\pi}{\alpha}\right) = \frac{\sin\left(\frac{p\pi}{\alpha}\right)}{\left(\frac{p\pi}{\alpha}\right)} = \frac{2 \tan(p\pi/2)}{p\pi \mathbf{E}(|X|^p)\mathbf{E}(|X|^{-p})}, \quad 0 < p < \min(\alpha, 1). \quad (2.30)$$

The above equation does not involve γ . Once α is estimated, γ can be obtained from (2.26):

$$\gamma = \left(\frac{\mathbf{E}(|X|^p)}{C_1(p, \alpha)} \right)^{\alpha/p}. \quad (2.31)$$

• **log |S α S| Estimator:**

By rewriting $\mathbf{E}(|X|^p)$ as $\mathbf{E}(e^{p \log |X|})$ and by defining the new random variable $Y = \log |X|$,* we have that:

$$\mathbf{E}(|X|^p) = \mathbf{E}(e^{p \log |X|}) = \mathbf{E}(e^{pY}), \quad -1 < p < \alpha, \quad (2.32)$$

where $\mathbf{E}(e^{pY})$ is the moment-generating function of Y . By expanding $\mathbf{E}(e^{pY})$ into power series:

$$\mathbf{E}(e^{pY}) = \sum_{k=0}^{\infty} \mathbf{E}(Y^k) \frac{p^k}{k!}, \quad (2.33)$$

and by using with (2.26), we can see that the moments of Y of any order must be finite and they satisfy:

$$\mathbf{E}(Y^k) = \frac{d^k}{dp^k} \left(C_1(p, \alpha) \gamma^{p/\alpha} \right) \Big|_{p=0}. \quad (2.34)$$

Simplifying the above equation, we have [18]:

$$\mathbf{E}(Y) = C_e \left(\frac{1}{\alpha} - 1 \right) + \frac{1}{\alpha} \log \gamma, \quad (2.35)$$

where $C_e = 0.57721566 \dots$ is the Euler constant, α is the characteristic exponent, γ is the dispersion, and

$$\text{Var}(Y) = \mathbf{E}\{(Y - \mathbf{E}\{Y\})^2\} = \frac{\pi^2}{6} \left(\frac{1}{\alpha^2} + \frac{1}{2} \right), \quad (2.36)$$

$$\mathbf{E}\{(Y - \mathbf{E}\{Y\})^3\} = 2\zeta(3) \left(\frac{1}{\alpha^3} - 1 \right), \quad (2.37)$$

where $\zeta(\cdot)$ is the Riemann Zeta function, and $\zeta(3)$ is a constant: $\zeta(3) = 1.2020569 \dots$. The higher-order moments of Y always exist and starting from the second-order moment, the equations only involve α . This property provides a simple algorithm to estimate the parameters (α, γ) of a $S\alpha S$ random process. Since we can estimate the mean and variance of Y by:

$$\bar{Y} = \frac{\sum_{i=1}^N Y_i}{N}, \quad \hat{\sigma}_Y^2 = \frac{\sum_{i=1}^N (Y_i - \bar{Y})^2}{N-1}, \quad (2.38)$$

*Note that $\log |X|$ is bounded because the p.d.f. of X $f(x)$ is bounded at $x = 0$.

Table 2.1: Performance comparison of the $\log |S\alpha S|$ Estimator versus the Sinc Function Estimator. The sample size is 5000, the true values are $\alpha = 1.5$ and $\gamma = 1$.

Estimation Method	$\hat{\alpha}$	$\hat{\gamma}$
$\log S\alpha S $ Estimator	1.4969 (0.0522)	0.9989 (0.0385)
Sinc Function Estimator	1.5027 (0.0536)	1.0023 (0.0423)

where N is the number of i.i.d. samples of $\{Y_i\}$, by solving (2.36) we can obtain an estimate of α and substitute into (2.35) for an estimate of γ . The $\log |S\alpha S|$ process was first introduced by Zolotarev [5].

Table 2.1 shows the computer simulation results for the above two estimators. As we can see, both estimators have mean value close to the true one and low variances (numbers in parentheses.)

2.2.1 Mountaintop Data Analysis: Clutter Modeling Using $S\alpha S$ Theory

The data files used in our experiments are from the Mountaintop Data Package tape. The clutter data set contains measured monostatic clutter using the Radar Surveillance Technology Experimental Radar (RSTER). The data consist of four files: cm435al.bfr, cm435bl.bfr, cm435cl.bfr, and ncal435a.bfr. These data files were collected on August 27, 1993 as part of the Clutter Map Multi-Frequency Experiment. The data are unequalized I/Q data and the waveform employed was a 5 μ Sec pulsed CW signal at 435 MHz. file cm435al covers from 3.5 to 43.5 nmi in range. File cm435bl covers from 40 to 80 nmi in range. File cm435cl covers from 75 to 115 nmi in range. Each range has a sample size of 495. All of these files have enough CPIs to cover 360 degrees in azimuth. The file ncal435a contains the system noise. In our comprehensive study, we conduct our experiments in 24 different azimuth angles with 15 degrees interval. At each azimuth angle of interest, we concatenate the data from the three ranges to form a single long stream (1485 samples). Experiments with the actually raw data are performed with the I/Q clutter data on Sensor # 1 (There are 14 sensors in the linear array). For the $S\alpha S$ modeling, the parameters (characteristic exponent α and dispersion γ) are first estimated using fractional lower order moments, then estimated by amplitude probability density (APD) curve fitting using exhaustive search. Noting that these parameters should remain the same for both the I- and Q-components of the clutter data, in the analysis of APD we focus on the I-component for simplicity. In the Gaussian model, the variance is estimated by taking standard deviation of the data. The samples are assumed to be i.i.d. and for convenience, the data have been scaled by 10^5 . In Figures 2.4 to 2.17, we show the I/Q data time series, estimated parameters, and APD comparison along several azimuth angle values at 15, 45, 60, 105, 120, 240, and 315 degrees. The experiments demonstrate that the $S\alpha S$ distribution is superior to the traditional Gaussian distribution for modeling the actual radar clutter data.

2.3 Signal Detection in Sub-Gaussian Impulsive Interference

In this section, we address the problem of coherent detection of a signal embedded in heavy-tailed noise modeled as a sub-Gaussian, alpha-stable process. We assume that the signal is a complex-valued vector of length L , known only within a multiplicative constant. The dependence structure of the noise, i.e., the underlying matrix of the sub-Gaussian process, is not known. The intent is to implement a generalized likelihood ratio detector which employs robust estimates of the unknown noise underlying matrix and the unknown signal strength. The performance of the proposed adaptive detector is compared to that of an adaptive matched filter that uses Gaussian estimates of the noise underlying matrix and the signal strength and is found to be clearly superior. The proposed new algorithms are theoretically analyzed and illustrated in a Monte-Carlo simulation.

2.3.1 Estimation of the Underlying Matrix of a Sub-Gaussian Vector

Before we attack the signal detection problem in sub-Gaussian interference, we present some necessary theory in the estimation of the underlying matrix of a sub-Gaussian vector. The following proposition found in [10, pp. 89] expresses the underlying matrix of a sub-Gaussian vector in terms of its covariation matrix and can, therefore, be used to obtain high quality estimates of the underlying matrix of the vector from independent observations.

Proposition 2.2 *Let $\underline{X} = [X_1, X_2, \dots, X_L]^T$ be a sub-Gaussian random vector with underlying covariance function \underline{R} (cf. (2.6)). Then, its covariation matrix \underline{C} will consist of the elements*

$$C_{ij} \equiv [X_i, X_j]_\alpha = 2^{-\frac{2}{\alpha}} R_{ij} R_{jj}^{\frac{\alpha-2}{2}} \quad (2.39)$$

The usefulness of the proposition lies in finding consistent estimators of the underlying covariance matrix of a sub-Gaussian vector from independent realizations $\underline{X}^1, \underline{X}^2, \dots, \underline{X}^K$ of the vector. The elements C_{ij} of the covariation matrix \underline{C} can be estimated by combining moment approximations of the expressions (2.14), (2.15), and (2.16) as follows

$$\hat{C}_{ij} = C(p, \alpha) \left[\frac{1}{K} \sum_{k=1}^K X_i^k (X_j^k)^{<p-1>} \right]^{\alpha/p} \left[\frac{1}{K} \sum_{k=1}^K |X_j^k|^p \right]^{\alpha/p-1}, \quad (2.40)$$

where any $p < \frac{\alpha}{2}$ will result in a consistent estimate[†] [19], as shown in the following proposition.

Proposition 2.3 *The estimator \hat{C}_{ij} of the covariation matrix elements given in (2.40), where $p < \alpha/2$, is consistent and asymptotically normal with mean C_{ij} and covariance $\mathcal{E}\{(\hat{C}_{ij} - C_{ij})(\hat{C}_{lm} - C_{lm})^*\}$ as in Appendix A.*

Proof See Appendix A. ■

Equation (2.39) can now be used to compute an estimate of the underlying matrix \underline{R} from the estimate of the covariation matrix \underline{C} .

[†]We have empirically found that a good choice is $p = \frac{\alpha}{3}$.

Proposition 2.4 Let \hat{C}_{ij} be the estimator of the covariation matrix elements given in (2.40), where $p < \alpha/2$. The estimates

$$\hat{R}_{jj} = [2^{\frac{\alpha}{2}} \hat{C}_{jj}]^{\frac{2}{\alpha}} \quad (2.41)$$

$$\hat{R}_{ij} = 2^{\frac{\alpha}{2}} \hat{C}_{ij} / \hat{R}_{jj}^{\frac{\alpha-2}{2}} \quad (2.42)$$

of the elements of the underlying covariance matrix are consistent and asymptotically normal with means R_{jj} and R_{ij} , respectively, and variances as in Appendix B.

Proof See Appendix B. ■

The procedure is illustrated with the following simulation study: Consider a sub-Gaussian random vector of length $L = 64$ and underlying matrix $\underline{R} = \text{diag}\{1, 1, \dots, 1\}$. We assume that $K = 500$ independent realizations of the vector are available and plot the 32nd row of the mean over 100 Monte-Carlo simulations of the following two estimates:

$$\hat{\underline{R}} = \frac{1}{K} \sum_{k=1}^K \underline{X}^k \underline{X}^{kT} \quad (2.43)$$

$$\hat{\underline{R}} = \text{as obtained from covariation matrix estimates (2.42)}. \quad (2.44)$$

We examined the cases of $\alpha = 2$ and $\alpha = 1.5$. Clearly, the Gaussian estimate fails when $\alpha = 2$, while the covariation-based estimate maintains high performance in both the cases of $\alpha = 2$ and $\alpha = 1.5$.

Next, we consider the estimation of the amplitude of a signal of known shape embedded in sub-Gaussian noise from a number of independent observations. The following Proposition outlines the procedure and states its performance.

Proposition 2.5 Consider the collection of K vectors $\underline{X}^k = A\underline{s} + \underline{N}^k$, $k = 1, 2, \dots, K$, where $\underline{s}^T \underline{s} = 1$. Form the least-squares estimates $\hat{A}_k = \underline{s}^T \underline{X}^k = \underline{s}^T A \underline{s} + \underline{s}^T \underline{N}^k = A + \underline{s}^T \underline{N}^k$, $k = 1, 2, \dots, K$. Define $\hat{A} = \text{sm}\{\hat{A}_1, \hat{A}_2, \dots, \hat{A}_K\}$, where $\text{sm}\{\dots\}$ indicates the sample median of its arguments. The estimate \hat{A} is consistent and asymptotically normal with mean equal to the true signal amplitude A and variance $\frac{1}{K} [\frac{\pi \alpha \gamma^{1/\alpha}}{2\Gamma(1/\alpha)}]^2$, where $\gamma = 2^{-\frac{2}{\alpha}} (\sum_{i=1}^L \sum_{j=1}^L s_i s_j^* R_{ij})^{\frac{\alpha}{2}} = 2^{-\frac{2}{\alpha}} (\underline{s}^T \underline{R} \underline{s})^{\frac{\alpha}{2}}$.

Proof The random variables \hat{A}_k , $k = 1, 2, \dots, K$, will be independent, each of pdf $f(x)$, which can be computed as follows. We begin with

$$\hat{A}_k = A + \underline{s}^T \underline{N}^k = A + w^{\frac{1}{2}} \underline{s}^T \underline{G}^k,$$

where \underline{G}^k , $k = 1, 2, \dots, K$, are independent Gaussian random vectors, each of mean zero and covariance matrix \underline{R} . Therefore, $\underline{s}^T \underline{G}^k$, $k = 1, 2, \dots, K$, are independent Gaussian random variables of mean zero and variance $\underline{s}^T \underline{R} \underline{s}$, which implies that $\underline{s}^T \underline{N}^k$, $k = 1, 2, \dots, K$, are independent sub-Gaussian random variables of length $L = 1$ and dispersion $\gamma = 2^{-\frac{2}{\alpha}} (\underline{s}^T \underline{R} \underline{s})^{\frac{\alpha}{2}}$. Thus, $f(x) = f_{\alpha}(\gamma, A; x)$.

From [20, p. 369], it follows that the sample median of \hat{A}_k , $k = 1, 2, \dots, K$, is asymptotically (for $K \rightarrow \infty$) normal with mean equal to the true median (A) and variance $\frac{1}{K}[\frac{1}{2f_\alpha(\gamma, A; A)}]^2$. But, $f_\alpha(\gamma, \delta; x) = \gamma^{-1/\alpha} f_\alpha[1, 0; (x - \delta)\gamma^{-1/\alpha}]$ and $f_\alpha(1, 0; 0) = \frac{1}{\pi\alpha}\Gamma(\frac{1}{\alpha})$ [21]. Combining the last two relations, we get

$$\frac{1}{K}[\frac{1}{2f_\alpha(\gamma, A; A)}]^2 = \frac{1}{K}[\frac{\pi\alpha\gamma^{1/\alpha}}{2\Gamma(1/\alpha)}]^2$$

as the asymptotic variance of the estimator \hat{A} . ■

2.3.2 Data-Adaptive Algorithms for Coherent Signal Detection

We consider the hypothesis testing problem

$$\begin{aligned} H_0 &: \underline{X}^k = \underline{N}^k \\ & \quad k = 1, 2, \dots, K \\ H_1 &: \underline{X} = \underline{S} + \underline{N}^k, \end{aligned}$$

where all the vectors have dimension (length) L and $k = 1, 2, \dots, K$ indexes independent, identically distributed realizations.

We make the following assumptions:

1. The noise vectors \underline{N}^k have a sub-Gaussian distribution, i.e.,

$$\underline{N}^k = w_k^{\frac{1}{2}} \underline{G}^k,$$

where w_k is a positive $(\alpha/2)$ -stable random variable of unit dispersion, \underline{G} is a Gaussian random vector of covariance matrix \underline{R} , and w and \underline{G} are independent.

2. The signal vector $\underline{S} = A\underline{s}$ consists of a known shape \underline{s} (for which $\underline{s}^T \underline{s} = 1$) and an unknown amplitude A .

The proposed test statistic is a generalized likelihood ratio test that makes use of the multi-dimensional Cauchy pdf defined in (2.8):

$$t_C = \sum_{k=1}^K \log \left[\frac{1 + \underline{X}^T \hat{\underline{R}}^{-1} \underline{X}}{1 + (\underline{X} - \hat{A}\underline{s})^T \hat{\underline{R}}^{-1} (\underline{X} - \hat{A}\underline{s})} \right] \quad (2.45)$$

For the estimates $\hat{\underline{R}}$ and \hat{A} , we choose the estimates proposed in (2.42) and Proposition 2.5, respectively. Assuming Gaussian noise of unknown covariance matrix \underline{R} and unknown signal amplitude, the data-adaptive detector attains the form of an adaptive matched-filter, i.e., it computes the test statistic

$$t_G = (2/K) \sum_{k=1}^K (\hat{A}\underline{s})^T \hat{\underline{R}}^{-1} \underline{X} - |\hat{A}|^2 \underline{s}^T \hat{\underline{R}}^{-1} \underline{s}, \quad (2.46)$$

where $\hat{A} = (1/K) \sum_{k=1}^K \frac{\underline{s}^T \hat{\underline{R}}^{-1} \underline{X}^k}{\underline{s}^T \hat{\underline{R}}^{-1} \underline{s}}$ and $\hat{\underline{R}} = (1/K) \sum_{k=1}^K (\underline{X} - \hat{A}\underline{s})(\underline{X} - \hat{A}\underline{s})^T$.

2.3.3 Performance Analysis and Computer Illustration

A. Theoretical Performance Analysis

The theoretical performance of the described detection algorithm is analyzed by means of the following propositions.

Proposition 2.6 *Assume that the signal amplitude A and the noise underlying matrix \underline{R} were known, while the test statistic*

$$t_C^c = \sum_{k=1}^K \log \left[\frac{1 + \underline{X}^T \underline{R}^{-1} \underline{X}}{1 + (\underline{X} - A\underline{s})^T \underline{R}^{-1} (\underline{X} - A\underline{s})} \right] \quad (2.47)$$

were used for detection purposes. The asymptotic (for $K \rightarrow \infty$) receiver operating characteristic would then be

$$P_d^c = \frac{1}{2} \operatorname{erfc} \left[\frac{\sqrt{2\sigma_{H_0}^2} \operatorname{erfc}^{-1}(2P_{fa}^c) + m_{H_0} - m_{H_1}}{\sqrt{2\sigma_{H_1}^2}} \right], \quad (2.48)$$

where P_d^c is the probability of detection induced by the test statistic t_C^c , when operating at a probability of false alarm P_{fa}^c . In (2.48), we have made use of the notation $m_{H_0} \equiv \mathcal{E}\{t_C^c|H_0\} < \infty$, $m_{H_1} \equiv \mathcal{E}\{t_C^c|H_1\} < \infty$, $\sigma_{H_0}^2 \equiv \operatorname{var}\{t_C^c|H_0\} < \infty$, $\sigma_{H_1}^2 \equiv \operatorname{var}\{t_C^c|H_1\} < \infty$.

Proof From the tail behavior of the sub-Gaussian distributions, we can show that the above-defined $m_{H_0}, m_{H_1}, \sigma_{H_0}^2, \sigma_{H_1}^2$ are finite. Therefore, as $K \rightarrow \infty$, the test statistic t_C^c approaches a Gaussian random variable under either hypothesis. From this observation, the asymptotic performance follows. ■

Proposition 2.7 *Let t_C^c be as in Proposition 2.6, (cf. (2.47)). The asymptotic (as $K \rightarrow \infty$) performance of the test statistic t_C in (2.45) is the same as the asymptotic performance of the test statistic t_C^c that requires knowledge of the signal amplitude and the noise underlying matrix.*

Proof From Propositions 2.4 and 2.5, $\hat{\underline{R}}$ and \hat{A} are consistent estimators of \underline{R} and A , respectively, as $K \rightarrow \infty$. Slutsky's theorem [22, p. 461] guarantees that the asymptotic distribution of t_C will be the same as the asymptotic distribution of t_C^c and subsequently t_C will result in the same asymptotic receiver operating characteristic as t_C^c in Proposition 2.5. ■

B. Computer Illustration of the Detector Performance

The small sample performance of both the Gaussian and the proposed Cauchy detectors can be accurately assessed only via Monte-Carlo simulation. To this end, we chose an observation vector of length $L = 8$ and $K = 10$ independent copies of it, while for the signal we chose a shape of a square pulse of height $1/\sqrt{L}$ and an amplitude of $A = 1$. The sub-Gaussian interference was assumed to be of characteristic exponent $\alpha = 2, 1.75, 1.5, 1.25, 1$, and 0.75 , and underlying matrix $\underline{R} = \operatorname{diag}\{1, 1, \dots, 1\}$. The performance of the Gaussian and the Cauchy detectors was assessed via 10,000 Monte-Carlo runs.

In Figure 2.19, we compare the performance of the Gaussian and the Cauchy detectors for different values of the characteristic exponent α . We see that, for $\alpha = 2$, the Gaussian detector, as expected, outperforms the Cauchy detector; however, for all other values of α , the Cauchy

detector maintains a high performance level, while the performance of the Gaussian detector deteriorates down to unacceptably low levels. In Figure 2.20, we show the performance of the Gaussian and the Cauchy detectors for different values of the characteristic exponent α .

2.3.4 The Special Case of a Single Observation

Consider the data model

$$\underline{X} = A\underline{s} + \underline{N}, \quad (2.49)$$

where A is an unknown signal amplitude, \underline{s} is a known signal shape, and \underline{N} is a sub-Gaussian random noise vector. Let us for the moment assume that the underlying matrix of the distribution is known. Several immediate observations become apparent: The maximum likelihood estimate \hat{A}_{ml} satisfies

$$\hat{A}_{ml} = \operatorname{argmax}_a f(\underline{X} - a\underline{s}), \quad (2.50)$$

where $f(\cdot)$ is the known sub-Gaussian distribution. We can make use of the representation (2.9) and proceed to write

$$\hat{A}_{ml} = \operatorname{argmax}_a \int_0^\infty f_1(u) f_2(\underline{X} - a\underline{s}|u) du, \quad (2.51)$$

where $f_1(\cdot)$ is the distribution of the square root of a positive, $\frac{\alpha}{2}$ -stable random variable and $f_2(\underline{X} - a\underline{s}|u) = f_G(\frac{\underline{X} - a\underline{s}}{u})$ with $f_G(\cdot)$ the multivariate Gaussian distribution of covariance matrix \underline{R} . Clearly:

$$\hat{A}_{ml} = \operatorname{argmax}_a f_G\left(\frac{\underline{X} - a\underline{s}}{u}\right), \quad (2.52)$$

i.e.,

$$\hat{A}_{ml} = \frac{\underline{s}^T \underline{R}^{-1} \underline{X}}{\underline{s}^T \underline{R}^{-1} \underline{s}}. \quad (2.53)$$

We have, in other words, shown that *the maximum likelihood estimate \hat{A}_{ml} is the least-squares estimate, as in the Gaussian case; however, in the sub-Gaussian case, the estimate is not efficient and, in fact, contains infinite error variance.* As a consequence, the estimate \hat{A}_{ml} is not consistent and the performance of the test statistic t_C in which this estimate is used will not approximate the performance of the test statistic t_C^e in which exact knowledge of A (and \underline{R}) is assumed.

2.3.5 Summary and Conclusions

In this section, we addressed the problem of detection of a signal, known within a multiplicative constant, in sub-Gaussian impulsive interference of unknown underlying matrix. We defined the sub-Gaussian class of random processes, summarized several key results, and presented data-adaptive algorithms for coherent detection of a known (within a multiplicative constant) signal. Our derivations were based on a solid mathematical basis.

From this study, we found that the Gaussian detectors for the same problem deteriorate in performance when required to operate in sub-Gaussian interference. On the other hand, a detector based on the multidimensional Cauchy distribution exhibited resistance to the presence of the sub-Gaussian interference and high performance, comparable to the performance of the Gaussian detector in Gaussian interference.

2.4 Target Angle and Doppler Estimation for Airborne Radar in Impulsive Interference Modeled as a Stable Process

In this section we develop target angle and Doppler, maximum likelihood-based estimation techniques from radar measurements retrieved in the presence of impulsive noise modeled as a multivariate $S\alpha S$ random process. We derive the Cramér-Rao bounds for the additive $S\alpha S$ interference scenario to assess the best-case estimation accuracy which can be achieved. In addition, we introduce a new joint spatial- and doppler-frequency high-resolution estimation technique based on the fractional lower-order statistics of the measurements of a radar array. We define the covariation matrix of the space-time radar observation vector process and employ subspace-based estimation techniques to the sample covariation matrix resulting in improved target-angle and Doppler estimates in the presence of impulsive interference. The results are of great importance in the study of space-time adaptive processing (STAP) for airborne pulse Doppler radar arrays operating in impulsive interference environments.

2.4.1 Maximum Likelihood Estimation in Alpha-Stable Noise: The Cauchy Beamformer

In this section, we develop the Maximum Likelihood (ML) estimator of the location and Doppler frequency of a target in the presence of noise modeled as a *complex isotropic α -stable process*. Initially, we concentrate on the additive complex Cauchy noise case. There are two reasons for doing this: First, the Cauchy distribution has a closed-form expression for its density function. This results in a straight-forward implementation of the maximum likelihood estimation, with closed form expressions for the Cramér-Rao bound. Secondly, it is shown through simulations that the Cauchy beamformer is very robust in different impulsive noise environments, i.e., its performance does not change significantly when the parameter α of the $S\alpha S$ noise varies in the interval $[1, 2]$.

A. Single Target Response Model

Consider a uniformly spaced linear array radar antenna consisting of N elements, which transmits a coherent burst of M pulses at a constant pulse repetition frequency (PRF) f_r and over a certain range of directions of interest. The pulse repetition interval is T_r ($f_r = 1/T_r$.) A space-time snapshot refers to the $MN \times 1$ vector of samples corresponding to a single range gate. Given a target at angle ϕ and Doppler frequency f , the space-time snapshot can be written as [23]

$$\mathbf{x} = \beta \mathbf{v}(\phi, f) + \mathbf{n}, \quad (2.54)$$

where β is the target's complex amplitude given by

$$\beta = x + jy. \quad (2.55)$$

The vector \mathbf{v} is an $NM \times 1$ vector called the *space-time steering vector*. It may be expressed as

$$\mathbf{v}(\phi, f) = \mathbf{b}(f) \otimes \mathbf{a}(\phi) \quad (2.56)$$

where $\mathbf{a}(\phi)$ is the $N \times 1$ *spatial steering vector* containing the interelement phase shifts for a target at ϕ , and $\mathbf{b}(f)$ is the $M \times 1$ *temporal steering vector* that contains the interpulse phase

shifts for a target with Doppler f . It is assumed that the functional form of $\mathbf{v}(\phi, f)$ is known. For a linear radar array whose elements are spaced a half-wavelength apart, the following two vectors are defined:

- Spatial steering vector: $\mathbf{a}(\psi) = [1, e^{j\psi}, \dots, e^{j(N-1)\psi}]^T$, where $\psi = \frac{2\pi d}{\lambda_0} \cos(\phi)$ is the *normalized spatial frequency* of the target, and
- Temporal steering vector: $\mathbf{b}(\omega) = [1, e^{j\omega}, \dots, e^{j(M-1)\omega}]^T$, where $\omega = \frac{2\pi f}{f_r}$ is the normalized Doppler frequency of the target.

The snapshot contains a noise component \mathbf{n} which includes clutter, jamming, thermal noise, and any other undesired signals. We model \mathbf{n} as a multivariate Cauchy process with pdf given by (cf. (2.8))

$$f(\mathbf{n}) = \frac{c \|\underline{R}\|^{-1/2}}{[1 + \mathbf{n}^T \underline{R}^{-1} \mathbf{n}]^{(MN+1)/2}}, \quad (2.57)$$

where \underline{R} is a positive-definite matrix which models the statistical dependence of the impulsive noise process, and $c = \frac{1}{\pi^{(MN+1)/2}} \Gamma(\frac{MN+1}{2})$. As a first approximation to the problem, we will assume that the noise present at the array is statistically independent both along the array sensors and along time. In this case, each component of the noise vector is modeled as a complex isotropic Cauchy process with marginal pdf given by

$$\chi_\gamma(r) = \frac{\gamma}{2\pi(r^2 + \gamma^2)^{3/2}}, \quad (2.58)$$

where γ is the noise dispersion. Under the independence assumption, it follows from (2.54) and (2.58) that the joint density function for the case of a single snapshot is given by [24]

$$f(\mathbf{n}) = \prod_{i=1}^{MN} f(n_i) = \frac{\gamma^{MN}}{(2\pi)^{MN} \prod_{i=1}^{MN} (\gamma^2 + |x_i - \beta v_i|^2)^{3/2}}, \quad (2.59)$$

where $v_i(\psi, \omega)$ is the i th component of the space-time steering vector $\mathbf{v}(\psi, \omega)$

The estimation problem involves four real valued parameters which we can arrange to form a 4×1 parameter vector

$$\Theta = [\theta_1 \ \theta_2 \ \theta_3 \ \theta_4] = [\psi \ \omega \ x \ y]. \quad (2.60)$$

Given a single snapshot \mathbf{x} and ignoring the constant terms, the likelihood function $L(\Theta)$, is given by

$$L(\Theta) = -\frac{3}{2} \sum_{i=1}^{MN} \log(\gamma^2 + |x_i - \beta v_i(\Theta)|^2) \quad (2.61)$$

B. Cramér-Rao Bound Analysis

The Cramér-Rao bound (CRB) for the error variance of an unbiased estimator $\hat{\Theta}$ is given by

$$E\{(\hat{\Theta} - \Theta)(\hat{\Theta} - \Theta)^T\} \geq \mathbf{J}^{-1}(\Theta), \quad (2.62)$$

where $\mathbf{J}(\Theta)$ is the Fisher information matrix given by

$$\mathbf{J}(\Theta) = E\{[\partial L(\Theta)/\partial \Theta][\partial L(\Theta)/\partial \Theta]^T\}. \quad (2.63)$$

Under the assumptions stated in Section 2.4.1, and for the case of complex isotropic Cauchy noise, the following theorem holds:

Theorem 2.1 *The CRB on the error variance of the target angle ϕ and Doppler frequency f estimates is given by*

$$CRB(\phi) = \frac{\gamma^2}{|\beta|^2} \frac{\lambda^2}{(2\pi d)^2} \cdot \frac{5N(\|\mathbf{d}^b\|^2 - \delta_b^2/M)}{3\xi} \cdot \frac{1}{\cos^2(\phi)}, \quad (2.64)$$

and

$$CRB(f) = \frac{\gamma^2}{|\beta|^2} \frac{1}{(2\pi T_r)^2} \cdot \frac{5M(\|\mathbf{d}^a\|^2 - \delta_a^2/N)}{3\xi}, \quad (2.65)$$

where

$$\begin{aligned} \mathbf{d}^a &= \partial \mathbf{a} / \partial \phi, & \|\mathbf{d}^a\|^2 &= \sum_{i=1}^N |\partial a_i / \partial \phi|^2, \\ \mathbf{d}^b &= \partial \mathbf{b} / \partial f, & \|\mathbf{d}^b\|^2 &= \sum_{i=1}^M |\partial b_i / \partial \omega|^2, \\ \delta_a &= \sum_{i=1}^N |d_i^a|, & \delta_b &= \sum_{i=1}^M |d_i^b|, & \rho &= \sum_{i=1}^{MN} |d_{g(i)}^a| |d_{f(i)}^b|, \end{aligned}$$

and

$$\xi = (M \|\mathbf{d}^a\|^2 - \frac{M}{N} \delta_a^2)(N \|\mathbf{d}^b\|^2 - \frac{N}{M} \delta_b^2) - (\delta_a \delta_b - \rho)^2.$$

Proof See Appendix C. ■

We should note that the above bounds can be achieved only when there exist unbiased estimators of all the model parameters. A useful insight on the CRB can be gained if we consider the case of a single source impinging in a linear array whose sensors are spaced a half-wavelength apart. In this case,

$$CRB(\phi) = \frac{\gamma^2}{|\beta|^2} \frac{\lambda^2}{(2\pi d)^2} \cdot \frac{20}{M^2 N^2 (N^2 - 1)} \cdot \frac{1}{\cos^2(\phi)}, \quad (2.66)$$

and

$$CRB(f) = \frac{\gamma^2}{|\beta|^2} \frac{1}{(2\pi T_r)^2} \cdot \frac{20}{N^2 M^2 (M^2 - 1)}. \quad (2.67)$$

The term $\gamma^2/|\beta|^2$ in the above expressions for the CRB can be viewed as the inverse of a quantity analogous to the signal-to-noise ratio (SNR) for the Gaussian case, i.e., a generalized SNR, so to speak. The larger the dispersion γ of the noise, the higher the CRB.

C. A Performance Study

To demonstrate the performance of the proposed method for the target parameter estimation problem, we conducted several simulation experiments where we compared the ML estimator based on the Cauchy noise assumption (MLC) with the ML estimator based on the Gaussian noise assumption (MLG), and with the MUSIC estimator.

In all the experiments the array is linear with five sensors spaced a half-wavelength apart. A single signal impinges to the array from a stationary target located at a direction of 5° . At first, the signal is assumed to be known at the receiver and the ML method is applied to estimate the

Table 2.2: GSNR and average PSNR for different values of M .

	Number of snapshots, M				
	$M = 5$	$M = 10$	$M = 20$	$M = 50$	$M = 100$
<i>GSNR</i> [dB]	22.6951	22.7323	22.7416	22.5003	22.6543
<i>PSNR</i> [dB]	7.3712	3.3938	0.5371	-4.3158	-7.149

direction of arrival. Also, for the unknown signal case the pseudo ML method is applied by using a least-squares estimate for the signal. The noise is assumed to follow the bivariate isotropic stable distribution.

In every experiment we perform 500 Monte-Carlo runs and compute the mean square error (MSE) of the direction-of arrival (DOA) estimates. The optimization for the MLC and MLG methods is performed by a steepest-descent algorithm with variable stepsize selected by means of Armijo's rule [25]. Since the alpha-stable family for $\alpha < 2$ determines processes with infinite variance, we define two alternative signal-to-noise ratios (SNR's). Namely, we define the *Generalized SNR* (*GSNR*) to be the ratio of the signal power over the noise dispersion γ :

$$GSNR = 10 \log\left(\frac{1}{\gamma M} \sum_{t=1}^M |s(t)|^2\right). \quad (2.68)$$

Also, for finite sample realizations, we define the *Pseudo-SNR* (*PSNR*) as:

$$PSNR = 10 \log\left(\frac{\sum_{t=1}^M |s(t)|^2}{\sum_{t=1}^M |n(t)|^2}\right). \quad (2.69)$$

In this example we study the estimation accuracy of MLC, MLG and MUSIC as a function of three parameters, namely the number of snapshots M , the noise dispersion γ , and the noise characteristic exponent α .

Number of snapshots M . In the first experiment we study the influence of the number of snapshots M to the performance of the algorithms. The noise follows the complex isotropic Cauchy distribution with dispersion $\gamma = 1$. For this experiment, the GSNR is kept almost constant at 22.5 dB as shown on Table 2.2. The PSNR is different in every Monte-Carlo run, so we calculate the average PSNR over the 500 Monte-Carlo runs (cf. Table 2.2). As the number of snapshots M increases, the PSNR decreases because more and more impulsive noise samples are incorporated into the data.

Figure 2.21(a) shows the resulting MSE of the estimated DOA as a function of the number of snapshots when the signal is known. The CRB is also plotted. As expected, the MLC estimator has the best performance since it is the optimal estimator for this type of noise. Also, the complete failure of the MLC and MUSIC processors for this type of impulsive noise is apparent. Figure 2.21(b) shows similar plots for the case of the pseudo ML estimators where we use a LS estimate for the signal. The MLC estimator has again the least MSE. Comparing these curves with the analogous curves obtained assuming exact signal knowledge, we observe a larger MSE for the pseudo ML estimates, as expected.

Noise dispersion γ . In the second experiment we study the influence of the noise dispersion γ , i.e., the influence of the GSNR to the performance of the methods. Here also, the noise follows

Table 2.3: GSNR and average PSNR for different values of γ .

	<i>Noise Dispersion, γ</i>						
	$\gamma = 0.5$	$\gamma = 1$	$\gamma = 2$	$\gamma = 4$	$\gamma = 6$	$\gamma = 8$	$\gamma = 10$
<i>GSNR [dB]</i>	25.7519	22.7416	19.7313	16.7210	14.9601	13.7107	12.7416
<i>PSNR [dB]</i>	6.5577	0.5371	-5.4835	-11.5041	-15.0259	-17.5247	-19.4629

Table 2.4: GSNR and average PSNR for different values of α .

	<i>Noise Characteristic Exponent, α</i>					
	$\alpha = 1.0$	$\alpha = 1.2$	$\alpha = 1.4$	$\alpha = 1.6$	$\alpha = 1.8$	$\alpha = 2.0$
<i>GSNR [dB]</i>	22.7416 ($\gamma = 1$)					
<i>PSNR [dB]</i>	0.5371	6.2171	10.1869	13.1035	15.3269	20.0383

the bivariate isotropic Cauchy distribution with dispersion γ . The number of snapshots available to the algorithms is $M = 20$. The GSNR and average PSNR for this experiment are shown on Table 2.3.

Figure 2.22 shows the resulting MSE of the estimated DOA as a function of the GSNR. Again the MLC estimate has the best performance. As evident in Figure 2.22(b), the performance of the MLC using a LS estimate for the signal degrades more rapidly for large values of the noise dispersion, γ (low GSNR values).

Characteristic exponent α . The importance of this experiment rests in its study of the robustness of the algorithms in different noise environments. Of course, by design, the MLG estimator is optimal for additive Gaussian noise ($\alpha = 2$), and the introduced MLC estimator is optimal for additive Cauchy noise ($\alpha = 1$). An important property of any processor is to be able to perform reasonably well in a wide range of noise environments ($1 < \alpha < 2$). Here, we test the performance of the estimators when the characteristic exponent, α , of the noise stable law is changing.

Figure 2.23 shows the resulting MSE curves as functions of the characteristic exponent α . The number of snapshots available to the algorithms is $M = 20$. The GSNR is 22.7416 dB ($\gamma = 1$) and is shown together with the average PSNR, on Table 2.4. The CRB as a function of α was computed as shown in [24].

As we can clearly see, the Cauchy beamformer is practically insensitive to the changes of α , and for exact signal knowledge it almost achieves the CRB for the whole range of values α . On the other hand, both the MLG and the MUSIC algorithms exhibit very large mean-square estimation errors for non-Gaussian noise environments. Note that when $\alpha = 2$, i.e., for the Gaussian noise case, the MLG method has the least MSE, as expected.

The experiment demonstrates that for values of α in [1,2], the ML method based on the Cauchy noise assumption exhibits performance very close to the optimum. This observation, combined with the fact that the ML method based on the Cauchy assumption has computational complexity similar to the ML method based on the Gaussian assumption, justify the importance of the Cauchy beamformer for the DOA estimation problem in practice.

D. Concluding Remarks

We considered the problem of target-angle and Doppler estimation with an airborne radar employing space-time adaptive processing. We derived Cramér-Rao bounds on angle and Doppler estimator accuracy for the case of additive multivariate Cauchy interference of known underlying matrix. The bounds are functions of a generalized SNR function, similarly to the Gaussian case where the bounds are functions of the SNR. The proposed method is based on the maximum likelihood estimation technique where the noise is modeled as a complex isotropic $S\alpha S$ process. The Cauchy beamformer has been shown to give better bearing estimates than the Gaussian beamformer in a wide range of impulsive noise environments, and for very low SNR values.

2.4.2 Subspace-Based STAP Using Lower-Order Statistics

In this section, we present a new subspace-based method for joint spatial- and doppler-frequency high-resolution estimation in the presence of impulsive noise which can be modeled as a complex symmetric alpha-stable ($S\alpha S$) process. We define the covariation matrix of the array sensor outputs and show that eigendecomposition-based methods, such as the MUSIC algorithm, can be applied to the sample covariation matrix to extract the doppler/angle information from the measurements. The improved performance of the proposed source localization method in the presence of a wide range of impulsive noise environments is demonstrated via Monte Carlo experiments.

A. STAP Problem Formulation

Consider a uniformly spaced linear array radar antenna consisting of N elements, which transmits a coherent burst of M pulses at a constant pulse repetition frequency (PRF) f_r and over a certain range of directions of interest. The array receives narrow-band signal returns reflected by q moving targets which are located at azimuth angles $\{\theta_k; k = 1, \dots, q\}$ and have relative velocities with respect to the radar $\{\nu_k; k = 1, \dots, q\}$ corresponding to Doppler frequencies $\{f_k; k = 1, \dots, q\}$. Since the signals are narrow-band, the propagation delay across the array is much smaller than the reciprocal of the signal bandwidth, and it follows that, by using a complex envelop representation, the array output can be expressed as [23]:

$$\mathbf{x}(t) = \mathbf{V}(\boldsymbol{\Theta}, \boldsymbol{\varpi})\mathbf{s}(t) + \mathbf{n}(t), \quad (2.70)$$

where

- $\mathbf{x}(t) = [x_1(t), \dots, x_{MN}(t)]^T$ is the array output vector (N : number of array elements, M : number of pulses, t may refer to the number of the coherent processing intervals (CPI's) available at the receiver);
- $\mathbf{s}(t) = [s_1(t), \dots, s_q(t)]^T$ is the signal vector emitted by the sources as received at the reference sensor 1 of the array;
- $\mathbf{V}(\boldsymbol{\Theta}, \boldsymbol{\varpi}) = [\mathbf{v}(\vartheta_1, \varpi_1), \dots, \mathbf{v}(\vartheta_q, \varpi_q)]$ is the *space-time steering matrix* ($\varpi_k = \frac{f_k}{f_r}$);
- Space-Time steering vector: $\mathbf{v}(\vartheta_k, \varpi_k) = \mathbf{b}(\varpi_k) \otimes \mathbf{a}(\vartheta_k)$;
 - $\mathbf{a}(\vartheta_k) = [1, e^{j2\pi\vartheta_k}, \dots, e^{j(N-1)2\pi\vartheta_k}]^T$ is the spatial steering vector ($\vartheta_k = \frac{d}{\lambda_0} \cos(\theta_k)$);
 - $\mathbf{b}(\varpi_k) = [1, e^{j2\pi\varpi_k}, \dots, e^{j(M-1)2\pi\varpi_k}]^T$ is the temporal steering vector.

- $\mathbf{n}(t) = [n_1(t), \dots, n_{MN}(t)]^T$ is the noise vector.

Assuming the availability of P CPI's t_1, \dots, t_P , the data can be expressed as

$$\mathbf{X} = \mathbf{V}(\boldsymbol{\Theta}, \boldsymbol{\varpi})\mathbf{S} + \mathbf{N}, \quad (2.71)$$

where \mathbf{X} and \mathbf{N} are the $MN \times P$ matrices

$$\mathbf{X} = [\mathbf{x}(t_1), \dots, \mathbf{x}(t_P)], \quad (2.72)$$

$$\mathbf{N} = [\mathbf{n}(t_1), \dots, \mathbf{n}(t_P)], \quad (2.73)$$

and \mathbf{S} is the $q \times P$ matrix

$$\mathbf{S} = [\mathbf{s}(t_1), \dots, \mathbf{s}(t_P)]. \quad (2.74)$$

Our objective is to jointly estimate the directions-of-arrival $\{\vartheta_k; k = 1, \dots, q\}$ and the Doppler frequencies $\{f_k; k = 1, \dots, q\}$ of the source targets.

B. The Array Covariation Matrix

We will assume that the q signal waveforms are non-coherent, statistically independent, complex isotropic $S\alpha S$ ($1 < \alpha \leq 2$) random processes with zero location parameter and covariation matrix $\boldsymbol{\Gamma}_S = \text{diag}(\gamma_{s_1}, \dots, \gamma_{s_q})$. Also, the noise vector $\mathbf{n}(t)$ is a complex isotropic $S\alpha S$ random process with the same characteristic exponent α as the signals. The noise is assumed to be independent of the signals with covariation matrix $\boldsymbol{\Gamma}_N = \gamma_n \mathbf{I}$.

Equation (2.70) can be written as

$$\mathbf{x}(t) = \mathbf{w}(t) + \mathbf{n}(t), \quad (2.75)$$

where $\mathbf{w}(t) = \mathbf{V}(\boldsymbol{\Theta}, \boldsymbol{\varpi})\mathbf{s}(t)$. By the stability property, it follows that $\mathbf{w}(t)$ is also a complex isotropic $S\alpha S$ random vector with components

$$w_i(t) = \mathbf{V}_i(\boldsymbol{\Theta}, \boldsymbol{\varpi})\mathbf{s}(t) = v_i(\vartheta_1, \varpi_1)s_1(t) + \dots + v_i(\vartheta_q, \varpi_q)s_q(t) \quad i = 1, \dots, MN. \quad (2.76)$$

Also, it holds that $\mathbf{w}(t)$ is independent of $\mathbf{n}(t)$.

Now, we define the *covariation matrix*, $\boldsymbol{\Gamma}_X$, of the observation vector process $\mathbf{x}(t)$ as the matrix whose elements are the covariations $[x_i(t), x_j(t)]_\alpha$ of the components of $\mathbf{x}(t)$. We have that

$$\begin{aligned} [x_i(t), x_j(t)]_\alpha &= [w_i(t) + n_i(t), w_j(t) + n_j(t)]_\alpha \\ &= [w_i(t), w_j(t)]_\alpha + [w_i(t), n_j(t)]_\alpha + [n_i(t), w_j(t)]_\alpha + [n_i(t), n_j(t)]_\alpha. \end{aligned} \quad (2.77)$$

By the independence assumption of $\mathbf{w}(t)$ and $\mathbf{n}(t)$ and by property P3 we have that

$$[w_i(t), n_j(t)]_\alpha = 0, \quad (2.78)$$

and

$$[n_i(t), w_j(t)]_\alpha = 0. \quad (2.79)$$

Also, by using (2.76) and properties P1 and P2 it follows that

$$\begin{aligned}
[w_i(t), w_j(t)]_\alpha &= \left[\sum_{k=1}^q v_i(\vartheta_k, \varpi_k) s_k(t), w_j(t) \right]_\alpha \\
&= \sum_{k=1}^q v_i(\vartheta_k, \varpi_k) [s_k(t), w_j(t)]_\alpha \\
&= \sum_{k=1}^q v_i(\vartheta_k, \varpi_k) \left[s_k(t), \sum_{l=1}^q v_j(\vartheta_l) s_l(t) \right]_\alpha \\
&= \sum_{k=1}^q v_i(\vartheta_k, \varpi_k) v_j^{<\alpha-1>}(\vartheta_k, \varpi_k) \gamma_{s_k}, \tag{2.80}
\end{aligned}$$

where $\gamma_{s_k} = [s_k, s_k]_\alpha$. Finally, due to the noise assumption made earlier, it holds that

$$[n_i(t), n_j(t)]_\alpha = \gamma_n \delta_{i,j}, \tag{2.81}$$

where $\delta_{i,j}$ is the Kronecker delta function. Combining (2.77)-(2.81) we obtain the following expression for the covariations of the sensor measurements:

$$[x_i(t), x_j(t)]_\alpha = \sum_{k=1}^q v_i(\vartheta_k, \varpi_k) v_j^{<\alpha-1>}(\vartheta_k, \varpi_k) \gamma_{s_k} + \gamma_n \delta_{i,j} \quad i, j = 1, \dots, MN. \tag{2.82}$$

In addition, the dispersion and covariation coefficients of the array sensor measurements are given respectively by

$$\gamma_{x_j(t)} = \sum_{k=1}^q |v_j(\vartheta_k, \varpi_k)|^\alpha \gamma_{s_k} + \gamma_n \quad j = 1, \dots, MN, \tag{2.83}$$

and

$$\lambda_{x_i(t), x_j(t)} = \frac{\sum_{k=1}^q v_i(\vartheta_k, \varpi_k) v_j^{<\alpha-1>}(\vartheta_k, \varpi_k) \gamma_{s_k} + \gamma_n \delta_{i,j}}{\sum_{k=1}^q |v_j(\vartheta_k, \varpi_k)|^\alpha \gamma_{s_k} + \gamma_n} \quad i, j = 1, \dots, MN. \tag{2.84}$$

In matrix form, (2.82) gives the following expression for the covariation matrix of the observation vector:

$$\Gamma_X \triangleq [\mathbf{x}(t), \mathbf{x}(t)]_\alpha = \mathbf{V}(\boldsymbol{\Theta}, \boldsymbol{\varpi}) \Gamma_S \mathbf{V}^{<\alpha-1>}(\boldsymbol{\Theta}, \boldsymbol{\varpi}) + \gamma_n \mathbf{I}, \tag{2.85}$$

where the (i, j) th element of matrix $\mathbf{V}^{<\alpha-1>}(\boldsymbol{\Theta}, \boldsymbol{\varpi})$ results from the (j, i) th element of $\mathbf{V}(\boldsymbol{\Theta}, \boldsymbol{\varpi})$ according to the operation

$$[\mathbf{V}^{<\alpha-1>}(\boldsymbol{\Theta}, \boldsymbol{\varpi})]_{i,j} = [\mathbf{V}(\boldsymbol{\Theta}, \boldsymbol{\varpi})]_{j,i}^{<\alpha-1>} = |[\mathbf{V}(\boldsymbol{\Theta}, \boldsymbol{\varpi})]_{j,i}|^{\alpha-2} [\mathbf{V}(\boldsymbol{\Theta}, \boldsymbol{\varpi})]_{j,i}^* \tag{2.86}$$

Clearly, when $\alpha = 2$, i.e., for Gaussian distributed signals and noise, the expression for the covariation matrix is identical to the well-known expression for the covariance matrix:

$$\mathbf{R}_X = \mathbf{V}(\boldsymbol{\Theta}, \boldsymbol{\varpi}) \Sigma \mathbf{V}^H(\boldsymbol{\Theta}, \boldsymbol{\varpi}) + \sigma^2 \mathbf{I}, \tag{2.87}$$

where Σ is the signal covariance matrix.

When the amplitude response of the sensors equals unity, it follows that

$$[\mathbf{V}^{<\alpha-1>}(\boldsymbol{\Theta}, \boldsymbol{\varpi})]_{i,j} = [\mathbf{V}(\boldsymbol{\Theta}, \boldsymbol{\varpi})]_{j,i}^*, \quad (2.88)$$

and thus the covariation matrix can be written as

$$\boldsymbol{\Gamma}_X = \mathbf{V}(\boldsymbol{\Theta}, \boldsymbol{\varpi}) \boldsymbol{\Gamma}_S \mathbf{V}^H(\boldsymbol{\Theta}, \boldsymbol{\varpi}) + \gamma_n \mathbf{I}. \quad (2.89)$$

Also, from (2.83) and (2.84) the dispersion and covariation coefficients of the array sensor measurements can be written as

$$\gamma_{x_j(t)} = \sum_{k=1}^q \gamma_{s_k} + \gamma_n \quad j = 1, \dots, MN, \quad (2.90)$$

and

$$\lambda_{x_i(t), x_j(t)} = \frac{\sum_{k=1}^q v_i(\vartheta_k, \varpi_k) v_j^*(\vartheta_k, \varpi_k) \gamma_{s_k} + \gamma_n \delta_{i,j}}{\sum_{k=1}^q \gamma_{s_k} + \gamma_n} \quad i, j = 1, \dots, MN. \quad (2.91)$$

Observing (2.89), we conclude that standard subspace techniques can be applied to the covariation or the covariation coefficient matrices of the observation vector to extract the bearing information. More specifically, it follows that the rank of matrix $\mathbf{V}(\boldsymbol{\Theta}, \boldsymbol{\varpi}) \boldsymbol{\Gamma}_S \mathbf{V}^H(\boldsymbol{\Theta}, \boldsymbol{\varpi})$ is q , with the smallest $(MN - q)$ of its eigenvalues equal to zero. In other words, if we let $\rho_1 \geq \rho_2 \geq \dots \geq \rho_{MN}$ denote the eigenvalues of matrix $\boldsymbol{\Gamma}_X$, then

$$\rho_{q+1} = \dots = \rho_{MN} = \gamma_n. \quad (2.92)$$

By denoting the corresponding eigenvectors of $\boldsymbol{\Gamma}_X$ by $\{\mathbf{u}_i\}_{i=1}^{MN}$, the ROC-MUSIC spectrum can be expressed as

$$S_{ROC-MUSIC}(\vartheta, \boldsymbol{\varpi}) = \frac{1}{\sum_{i=q+1}^{MN} |\mathbf{v}^H(\vartheta, \boldsymbol{\varpi}) \mathbf{u}_i|^2}. \quad (2.93)$$

The locations of the source signals are determined by the values of θ for which the spectrum given by (2.93) peaks.

In practice, we have to estimate the covariation matrix from a finite number of array sensor measurements. A proposed estimator for the covariation coefficient $\lambda_{x_i(t), x_j(t)}$ is called the *fractional lower order (FLOM) estimator* and is given by [26]

$$\hat{\lambda}_{x_i(t), x_j(t)} = \frac{\sum_{t=1}^n x_i(t) x_j^{<p-1>}(t)}{\sum_{t=1}^n |x_j(t)|^p} \quad (2.94)$$

for some $0 \leq p < \alpha$. We will refer to the new algorithm resulting from the eigendecomposition of the array covariation coefficient matrix as the **Robust Covariation-Based MUSIC** or **ROC-MUSIC**.

C. Simulations

In this section, we study the resolution capability and estimation accuracy of ROC-MUSIC versus MUSIC as functions of the noise characteristic exponent α , and the separation of the two sources. The array is linear with five sensors spaced half wavelength apart ($N = 5$). The number of transmitted pulses is $M = 10$. The noise follows the bivariate isotropic stable distribution.

Table 2.5: GSNR and average PSNR for different values of α .

	<i>Noise Characteristic Exponent α</i>					
	$\alpha = 1.01$	$\alpha = 1.2$	$\alpha = 1.4$	$\alpha = 1.6$	$\alpha = 1.8$	$\alpha = 2.0$
<i>GSNR [dB]</i>	22.3 ($\gamma = 1$)					
<i>PSNR [dB]</i>	-17.9690	-7.8245	-2.8001	-0.8380	-0.2164	-0.0441

Characteristic exponent α . In this experiment, we evaluate the performance of the two algorithms in a wide range of noise environments, from the more impulsive (α in the neighborhood of 1) to the Gaussian ones ($\alpha = 2$). The angles of arrival for the two signals are $\theta_1 = -5^\circ$ and $\theta_2 = 5^\circ$. The GSNR is 22.3 dB ($\gamma = 1$) and is shown together with the average PSNR on Table 2.5. The characteristic exponent α of the additive noise is unknown to the ROC-MUSIC algorithm. We use two values of the parameter p in the estimation of the covariation matrix (cf. (2.94)): $p = 0.8$ and $p = 0.4$. Clearly, MUSIC can be thought as a special case of ROC-MUSIC with $p = 2$.

In Figures 2.24-2.28, space-time spectral estimates obtained in five independent trials are shown for the ROC-MUSIC and MUSIC algorithms. Five types of alpha stable noise corresponding to three values of the characteristic exponent $\alpha = 1.1, 1.3, 1.5, 1.8, 2.0$ were used. We can see that the MUSIC method exhibits high-resolution performance only for the case of Gaussian additive noise. On the other hand, the ROC-MUSIC method exhibits better resolution capabilities for non-Gaussian additive noise environments.

Figure 2.29 depicts the improved performance of ROC-MUSIC over that of MUSIC both in terms of resolution probability and MSE, for values of α in the range (1, 2). Note that for $\alpha < 1.2$, MUSIC does not resolve the two sources in any of the 200 Monte Carlo runs. The results suggest that in impulsive noise environments modeled under the stable law, it is more beneficial to use the covariation matrix (lower-order moments) instead of the covariance matrix (second-order moments). Of course, for Gaussian additive noise ($\alpha = 2$) the use of second-order moments ($p = 2$) gives better results.

Angular separation. The resolution analysis of the two algorithms was studied by using a popular resolution criterion defined by the following threshold equation:

$$\Lambda(\theta_1, \theta_2) \triangleq P(\theta_m) - \frac{1}{2}\{P(\theta_1) + P(\theta_2)\} > 0, \quad (2.95)$$

where θ_1 and θ_2 are the angles of arrival of the two signals, $\theta_m = (\theta_1 + \theta_2)/2$ is the mid-range between them, and the *null spectrum* $P(\theta) \triangleq 1/S(\theta)$ is defined as the reciprocal of the spatial spectrum $S(\theta)$ given in (2.93). The two signals are said to be resolvable if inequality (2.95) holds. The inequality implies that the null spectrum magnitude at the mid-angle should lie above the line segment linking the two signal valleys, in order for the two sources to be resolvable. In our experiments, we estimated the null spectrum from a finite number of array sensor measurements and we determined the probability of resolution by averaging the resolution events over the 200 independent Monte Carlo runs.

Figure 2.30 illustrates the variation of the algorithmic performance with respect to the spatial angle separation of the two incoming signals for GSNR= 22.3 dB, $\text{PSNR}(\alpha = 1.5) = -1.56$ dB,

and $\text{PSNR}(\alpha = 1.8) = -0.22$ dB. As expected, the resolution capability of both algorithms improves with increased angle separation between the two sources. But for a given probability of resolution, the ROC-MUSIC algorithm requires a lower angle separation threshold than the MUSIC algorithm.

Appendix A: Proof of Proposition 2.3

Define

$$\begin{aligned} T_1 &= \frac{1}{K} \sum_{k=1}^K X_i^k (X_j^k)^{<p-1>} \\ T_2 &= \frac{1}{K} \sum_{k=1}^K |X_j^k|^p \\ T_3 &= \frac{1}{K} \sum_{k=1}^K X_m^k (X_n^k)^{<p-1>} \\ T_4 &= \frac{1}{K} \sum_{k=1}^K |X_n^k|^p. \end{aligned}$$

The vector $\begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \end{bmatrix}$ is asymptotically (as $K \rightarrow \infty$) normal with mean $\underline{m}_T = \begin{bmatrix} \mathcal{E}\{X_i(X_j)^{<p-1>}\} \\ \mathcal{E}\{|X_j|^p\} \\ \mathcal{E}\{X_m(X_n)^{<p-1>}\} \\ \mathcal{E}\{|X_n|^p\} \end{bmatrix}$

and correlation matrix $\underline{R}_T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & r_{14} \\ r_{21} & r_{22} & r_{23} & r_{24} \\ r_{31} & r_{32} & r_{33} & r_{34} \\ r_{41} & r_{42} & r_{43} & r_{44} \end{bmatrix}$, where

$$\begin{aligned} r_{11} &= \frac{1}{K} \mathcal{E}\{|X_i|^2 |X_j|^{2p-2}\} \\ r_{12} &= \frac{1}{K} \mathcal{E}\{X_i(X_j)^{<p-1>} |X_j|^p\} \\ r_{13} &= \frac{1}{K} \mathcal{E}\{X_i(X_j)^{<p-1>} X_m(X_n)^{<p-1>}\} \\ r_{14} &= \frac{1}{K} \mathcal{E}\{X_i(X_j)^{<p-1>} |X_n|^p\} \\ r_{21} &= r_{12}^* \\ r_{22} &= \frac{1}{K} \mathcal{E}\{|X_j|^{2p}\} \\ r_{23} &= \frac{1}{K} \mathcal{E}\{X_m(X_n)^{<p-1>} |X_j|^p\} \\ r_{24} &= \frac{1}{K} \mathcal{E}\{|X_n|^p |X_j|^p\} \\ r_{31} &= r_{13}^* \\ r_{32} &= r_{23}^* \\ r_{33} &= \frac{1}{K} \mathcal{E}\{|X_m|^2 |X_n|^{2p-2}\} \\ r_{34} &= \frac{1}{K} \mathcal{E}\{X_m(X_n)^{<p-1>} |X_n|^p\} \\ r_{41} &= r_{14}^* \\ r_{42} &= r_{24}^* \end{aligned}$$

$$\begin{aligned}
r_{43} &= r_{34}^* \\
r_{44} &= \frac{1}{K} \mathcal{E}\{|X_n|^{2p}\}.
\end{aligned}$$

Let

$$\begin{aligned}
g_1(u_1, u_2, u_3, u_4) &= C(p, \alpha) u_1^{\alpha/p} u_2^{\alpha/p-1} \\
g_2(u_1, u_2, u_3, u_4) &= C(p, \alpha) u_3^{\alpha/p} u_4^{\alpha/p-1}
\end{aligned}$$

and define $\underline{\underline{G}} = \begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \frac{\partial g_1}{\partial u_2} & \frac{\partial g_1}{\partial u_3} & \frac{\partial g_1}{\partial u_4} \\ \frac{\partial g_2}{\partial u_1} & \frac{\partial g_2}{\partial u_2} & \frac{\partial g_2}{\partial u_3} & \frac{\partial g_2}{\partial u_4} \end{bmatrix}$, where

$$\begin{aligned}
\frac{\partial g_1}{\partial u_1} &= C(p, \alpha) \frac{\alpha}{p} (u_1 u_2)^{\alpha/p-1} \\
\frac{\partial g_1}{\partial u_2} &= C(p, \alpha) \left(\frac{\alpha}{p} - 1\right) u_1^{\frac{\alpha}{p}} u_2^{\frac{\alpha}{p}-2} \\
\frac{\partial g_1}{\partial u_3} &= 0 \\
\frac{\partial g_1}{\partial u_4} &= 0 \\
\frac{\partial g_2}{\partial u_1} &= 0 \\
\frac{\partial g_2}{\partial u_2} &= 0 \\
\frac{\partial g_2}{\partial u_3} &= C(p, \alpha) \frac{\alpha}{p} (u_3 u_4)^{\alpha/p-1} \\
\frac{\partial g_2}{\partial u_4} &= C(p, \alpha) \left(\frac{\alpha}{p} - 1\right) u_3^{\frac{\alpha}{p}} u_4^{\frac{\alpha}{p}-2}
\end{aligned}$$

and $\underline{\underline{G}}$ is evaluated at $\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \underline{\underline{m}}_T$.

Then: $\begin{bmatrix} \hat{C}_{ij} \\ \hat{C}_{mn} \end{bmatrix} = \begin{bmatrix} g_1(T_1, T_2, T_3, T_4) \\ g_2(T_1, T_2, T_3, T_4) \end{bmatrix}$ will be asymptotically normal with mean $\begin{bmatrix} C_{ij} \\ C_{mn} \end{bmatrix}$ and covariance matrix $\underline{\underline{G}}(\underline{\underline{R}}_T - \underline{\underline{m}}_T \underline{\underline{m}}_T^T) \underline{\underline{G}}^T$. ■

Appendix B: Proof of Proposition 2.4

From the results in Appendix A and (2.42), we immediately see that \hat{R}_{jj} is asymptotically normal with mean R_{jj} and variance $\frac{1}{K} \sigma_{C_{jj}}^2 \left(\frac{4}{\alpha} C_{jj}^{2/\alpha-1}\right)^2$, where $\frac{1}{K} \sigma_{C_{jj}}^2$ is the asymptotic error variance in the estimate \hat{C}_{ij} .

Let now $h(u_1, u_2) = 2u_1 u_2^{\frac{2-\alpha}{\alpha}}$. Then:

$$\frac{\partial h}{\partial u_1} = 2u_2^{\frac{2-\alpha}{\alpha}}$$

$$\frac{\partial h}{\partial u_2} = 2u_1 \left(\frac{2-\alpha}{\alpha} \right) u_2^{\frac{2-2\alpha}{\alpha}}.$$

Therefore:

$$\sigma_{R_{ij}}^2 = [2C_{jj}^{*\frac{2-\alpha}{\alpha}}, 2(\frac{2-\alpha}{\alpha})C_{jj}^{*\frac{2-2\alpha}{\alpha}}] \begin{bmatrix} \sigma_{C_{ij}C_{ij}}^2 & \sigma_{C_{ij}C_{jj}}^2 \\ \sigma_{C_{ij}C_{jj}}^2 & \sigma_{C_{jj}C_{jj}}^2 \end{bmatrix} \begin{bmatrix} 2C_{jj}^{\frac{2-\alpha}{\alpha}} \\ 2(\frac{2-\alpha}{\alpha})C_{ij}C_{jj}^{\frac{2-2\alpha}{\alpha}} \end{bmatrix}.$$

Appendix C: Derivation of CRB for Complex Isotropic Cauchy Noise

The log-likelihood function is given by (2.61)

$$L(\Theta) = -\frac{3}{2} \sum_{i=1}^{NM} \log(\gamma^2 + |x_i - \beta v_i(\psi, \omega)|^2). \quad (2.96)$$

The derivatives of $L(\Theta)$ with respect to $[\psi, \omega, x, y]$, are as follows

$$\begin{aligned} \frac{\partial L}{\partial \psi} &= -\frac{3}{2} \sum_{i=1}^{MN} \frac{1}{\gamma^2 + |n_i|^2} \cdot \frac{\partial}{\partial \psi} [\gamma^2 + (x_i - \beta v_i)^*(x_i - \beta v_i)] \\ &= -\frac{3}{2} \sum_{i=1}^{MN} \frac{1}{\gamma^2 + |n_i|^2} \cdot [-\beta^* b_{f(i)}^* (d_{g(i)}^a)^* n_i - \beta b_{f(i)} (d_{g(i)}^a) n_i^*] \\ &= 3 \sum_{i=1}^{MN} \frac{\Re[\beta^* b_{f(i)}^* (d_{g(i)}^a)^* n_i]}{\gamma^2 + |n_i|^2} \end{aligned}$$

where $d_i^a = \partial a_i / \partial \psi$. Likewise, we obtain:

$$\frac{\partial L}{\partial \omega} = 3 \sum_{i=1}^{MN} \frac{\Re\{\beta^* a_{g(i)}^* (d_{f(i)}^b)^* n_i\}}{\gamma^2 + |n_i|^2} \quad (2.97)$$

where $d_i^b = \partial b_i / \partial \omega$. We also have that

$$\begin{aligned} \frac{\partial L}{\partial x} &= -\frac{3}{2} \sum_{i=1}^{MN} \frac{1}{\gamma^2 + |n_i|^2} \cdot [-a_{g(i)}^* b_{f(i)}^* n_i - a_{g(i)} b_{f(i)} n_i^*] \\ &= 3 \sum_{i=1}^{MN} \frac{\Re[a_{g(i)}^* b_{f(i)}^* n_i]}{\gamma^2 + |n_i|^2} \end{aligned}$$

and

$$\begin{aligned} \frac{\partial L}{\partial y} &= -\frac{3}{2} \sum_{i=1}^{MN} \frac{1}{\gamma^2 + |n_i|^2} \cdot [-j a_{g(i)}^* b_{f(i)}^* n_i - j a_{g(i)} b_{f(i)} n_i^*] \\ &= -3 \sum_{i=1}^{MN} \frac{\Im[a_{g(i)}^* b_{f(i)}^* n_i]}{\gamma^2 + |n_i|^2}. \end{aligned}$$

Now, we have to calculate the components of the Fisher information matrix.

$$E \left\{ \left(\frac{\partial L}{\partial \psi} \right)^2 \right\} = 9E \left\{ \sum_{i=1}^{MN} \sum_{j=1}^{MN} \frac{\Re[\beta^* b_{f(i)}^* (d_{g(i)}^a)^* n_i]}{\gamma^2 + |n_i|^2} \cdot \frac{\Re[\beta^* b_{f(j)}^* (d_{g(j)}^a)^* n_j]}{\gamma^2 + |n_j|^2} \right\}. \quad (2.98)$$

From the assumption made in Section 2.4.1, and observing that $\{|a_i| = 1, i = 1, 2, \dots, (N-1)\}$ and $\{|b_i| = 1, i = 1, 2, \dots, (M-1)\}$, we obtain:

$$\begin{aligned} E \left\{ \left(\frac{\partial L}{\partial \psi} \right)^2 \right\} &= 9E \left\{ \sum_{i=1}^{MN} \frac{(\Re[\beta^* b_{f(i)}^* (d_{g(i)}^a)^* n_i])^2}{(\gamma^2 + |n_i|^2)^2} \right\} \\ &= 9 \sum_{i=1}^{MN} E \left\{ \frac{(|\beta| |d_{g(i)}^a| |n_i| \cos(-\phi_\beta - \phi_{f(i)}^b - \phi_{g(i)}^a - \pi/2 + \phi_i^n))^2}{(\gamma^2 + |n_i|^2)^2} \right\}, \end{aligned}$$

where:

$$\phi_i^n = \arctan \left(\frac{\Im[n_i]}{\Re[n_i]} \right) \quad \phi_\beta = \arctan \left(\frac{y}{x} \right) \quad (2.99)$$

and

$$\phi_i^b = \arctan \left(\frac{\Im[b_i]}{\Re[b_i]} \right) \quad \phi_i^a = \arctan \left(\frac{\Im[a_i]}{\Re[a_i]} \right). \quad (2.100)$$

In addition,

$$E \left\{ \left(\frac{\partial L}{\partial \psi} \right)^2 \right\} = 9 \sum_{i=1}^{MN} \left[E \left\{ \frac{|n_i|^2}{(\gamma^2 + |n_i|^2)^2} \right\} |d_{g(i)}^a|^2 |\beta|^2 E \left\{ \cos^2(-\phi_\beta - \phi_{f(i)}^b - \phi_{g(i)}^a - \pi/2 + \phi_i^n) \right\} \right]. \quad (2.101)$$

Using the trigonometric propriety $\cos^2(\vartheta) = \frac{1}{2}(1 + \cos(2\vartheta))$, we obtain:

$$\begin{aligned} E \left\{ \left(\frac{\partial L}{\partial \psi} \right)^2 \right\} &= 9 \frac{2}{15\gamma^2} \sum_{i=1}^{MN} \left[|d_{g(i)}^a|^2 |\beta|^2 E \left\{ \frac{1}{2}(1 + \cos(-2\phi_\beta - 2\phi_{f(i)}^b - 2\phi_{g(i)}^a - \pi + 2\phi_i^n)) \right\} \right] \\ &= \frac{3}{5\gamma^2} |\beta|^2 \sum_{i=1}^{MN} |d_{g(i)}^a|^2 \\ &= \frac{3M}{5\gamma^2} |\beta|^2 \sum_{i=1}^N |d_i^a|^2. \end{aligned}$$

In the same way we can find:

$$E \left\{ \left(\frac{\partial L}{\partial \omega} \right)^2 \right\} = \frac{3N}{5\gamma^2} |\beta|^2 \sum_{i=1}^M |d_i^b|^2. \quad (2.102)$$

Now, we can obtain

$$\begin{aligned} E \left\{ \left(\frac{\partial L}{\partial x} \right)^2 \right\} &= 9 \sum_{i=1}^{MN} \sum_{j=1}^{MN} E \left\{ \frac{\Re[a_{g(i)}^* b_{f(i)}^* n_i]^2}{\gamma^2 + |n_i|^2} \cdot \frac{\Re[a_{g(j)}^* b_{f(j)}^* n_j]^2}{\gamma^2 + |n_j|^2} \right\} \\ &= 9 \sum_{i=1}^{MN} E \left\{ \frac{(\Re[a_{g(i)}^* b_{f(i)}^* n_i])^2}{(\gamma^2 + |n_i|^2)^2} \right\} \end{aligned}$$

$$\begin{aligned}
&= 9 \sum_{i=1}^{MN} E \left\{ \frac{|n_i|^2}{(\gamma^2 + |n_i|^2)^2} \right\} E \left\{ \frac{1}{2} + \frac{1}{2} \cos(-2\phi_{g(i)}^a - 2\phi_{f(i)}^b + 2\phi_i^n) \right\} \\
&= 9 \sum_{i=1}^{MN} \frac{2}{15\gamma^2} \frac{1}{2} = \frac{3}{5\gamma^2} MN.
\end{aligned}$$

Likewise we obtain:

$$E \left\{ \left(\frac{\partial L}{\partial y} \right)^2 \right\} = \frac{3}{5\gamma^2} MN. \quad (2.103)$$

Now:

$$\begin{aligned}
E \left\{ \frac{\partial L}{\partial \psi} \frac{\partial L}{\partial \omega} \right\} &= 9E \left\{ \sum_{i=1}^{MN} \sum_{j=1}^{MN} \frac{\Re[\beta^* b_{f(i)}^* (d_{g(i)}^a)^* n_i]}{\gamma^2 + |n_i|^2} \cdot \frac{\Re[\beta^* a_{g(j)}^* (d_{f(j)}^b)^* n_j]}{\gamma^2 + |n_j|^2} \right\} \\
&= 9 \sum_{i=1}^{MN} E \left\{ \frac{|\beta| |d_{g(i)}^a| |n_i| \cos(-\phi_\beta - \phi_{f(i)}^b - \phi_{g(i)}^a - \pi/2 + \phi_i^n)}{(\gamma^2 + |n_i|^2)} \right. \\
&\quad \left. \cdot \frac{|\beta| |d_{f(i)}^b| |n_i| \cos(-\phi_\beta - \phi_{f(i)}^b - \phi_{g(i)}^a - \pi/2 + \phi_i^n)}{(\gamma^2 + |n_i|^2)} \right\} \\
&= 9 \sum_{i=1}^{MN} \frac{2}{15\gamma^2} |d_{g(i)}^a| |d_{f(i)}^b| |\beta|^2 \frac{1}{2} = \frac{3}{5\gamma^2} |\beta|^2 \sum_{i=1}^{MN} [|d_{g(i)}^a| |d_{f(i)}^b|],
\end{aligned}$$

$$\begin{aligned}
E \left\{ \frac{\partial L}{\partial \psi} \frac{\partial L}{\partial x} \right\} &= 9E \left\{ \sum_{i=1}^{MN} \sum_{j=1}^{MN} \frac{\Re[\beta^* b_{f(i)}^* (d_{g(i)}^a)^* n_i]}{\gamma^2 + |n_i|^2} \cdot \frac{\Re[\beta^* a_{g(j)}^* b_{f(j)}^* n_j]}{\gamma^2 + |n_j|^2} \right\} \\
&= 9 \sum_{i=1}^{MN} E \left\{ \frac{|\beta| |d_{g(i)}^a| |n_i|^2 \cos(-\phi_\beta - \phi_{f(i)}^b - \phi_{g(i)}^a - \pi/2 + \phi_i^n) \cos(-\phi_{f(i)}^b - \phi_{g(i)}^a - \phi_i^n)}{(\gamma^2 + |n_i|^2)^2} \right\} \\
&= 9 \frac{2}{15\gamma^2} |\beta| \sum_{i=1}^{MN} |d_{g(i)}^a| \frac{1}{2} \cos(\phi_\beta + \frac{\pi}{2}) = \frac{3}{5\gamma^2} M y \sum_{i=1}^N |d_i^a|.
\end{aligned}$$

Likewise we obtain:

$$E \left\{ \frac{\partial L}{\partial \psi} \frac{\partial L}{\partial y} \right\} = \frac{3}{5\gamma^2} M x \sum_{i=1}^N |d_i^a|. \quad (2.104)$$

Now:

$$\begin{aligned}
E \left\{ \frac{\partial L}{\partial \omega} \frac{\partial L}{\partial x} \right\} &= 9E \left\{ \sum_{i=1}^{MN} \sum_{j=1}^{MN} \frac{\Re[\beta^* a_{g(i)}^* (d_{f(i)}^b)^* n_i]}{\gamma^2 + |n_i|^2} \cdot \frac{\Re[\beta^* a_{g(j)}^* b_{f(j)}^* n_j]}{\gamma^2 + |n_j|^2} \right\} \\
&= 9 \sum_{i=1}^{MN} E \left\{ \frac{|\beta| |d_{f(i)}^b| |n_i|^2 \cos(-\phi_\beta - \phi_{g(i)}^a - \phi_{f(i)}^b - \pi/2 + \phi_i^n) \cos(-\phi_{f(i)}^b - \phi_{g(i)}^a - \phi_i^n)}{(\gamma^2 + |n_i|^2)^2} \right\} \\
&= 9 \frac{2}{15\gamma^2} |\beta| \sum_{i=1}^{MN} |d_{f(i)}^b| \frac{1}{2} \cos(\phi_\beta + \frac{\pi}{2}) = \frac{3}{5\gamma^2} N y \sum_{i=1}^M |d_i^b|.
\end{aligned}$$

Likewise we obtain:

$$E \left\{ \frac{\partial L}{\partial \omega} \frac{\partial L}{\partial y} \right\} = \frac{3}{5\gamma^2} N x \sum_{i=1}^M |d_i^b|. \quad (2.105)$$

Finally,

$$\begin{aligned} E \left\{ \frac{\partial L}{\partial x} \frac{\partial L}{\partial y} \right\} &= 9E \left\{ \sum_{i=1}^{MN} \sum_{j=1}^{MN} \frac{\Re[a_{g(i)}^* b_{f(j)}^* n_i]}{\gamma^2 + |n_i|^2} \cdot \frac{\Im[a_{g(j)}^* b_{f(j)}^* n_j]}{\gamma^2 + |n_j|^2} \right\} \\ &= 9 \sum_{i=1}^{MN} E \left\{ \frac{|n_i|^2}{(\gamma^2 + |n_i|^2)^2} \cos(-\phi_{g(i)}^a - \phi_{f(i)}^b + \phi_i^n) \cdot \sin(-\phi_{f(i)}^b - \phi_{g(i)}^a - \phi_i^n) \right\} \\ &= 0. \end{aligned}$$

The Fisher Information matrix will be:

$$\mathbf{J}(\Theta) = \frac{3}{5\gamma^2} \cdot \begin{bmatrix} M|\beta|^2 \|\mathbf{d}^a\|^2 & |\beta|^2 \rho & yM\delta_a & xM\delta_a \\ |\beta|^2 \rho & N|\beta|^2 \|\mathbf{d}^b\|^2 & yN\delta_b & xN\delta_b \\ yM\delta_a & yN\delta_b & MN & 0 \\ xM\delta_a & xN\delta_b & 0 & MN \end{bmatrix}, \quad (2.106)$$

where $\|\mathbf{d}_a\|^2 = \sum_{i=1}^N |\partial a_i / \partial \psi|^2$ and $\|\mathbf{d}_b\|^2 = \sum_{i=1}^M |\partial b_i / \partial \omega|^2$, and:

$$\delta_a = \sum_{i=1}^N |d_i^a|, \quad (2.107)$$

$$\delta_b = \sum_{i=1}^M |d_i^b|, \quad (2.108)$$

and

$$\rho = \sum_{i=1}^{MN} |d_{g(i)}^a| |d_{f(i)}^b|. \quad (2.109)$$

Using the matrix inversion lemma [27] and after some simple matrix multiplication we have

$$\mathbf{J}_{2 \times 2}^{-1}(\Theta) = \frac{5\gamma^2}{3|\beta|^2} \cdot \begin{bmatrix} (N \|\mathbf{d}^b\|^2 - \frac{N}{M} \delta_b^2) & \delta_a \delta_b - \rho \\ \delta_a \delta_b - \rho & (M \|\mathbf{d}^a\|^2 - \frac{M}{N} \delta_a^2) \end{bmatrix} \cdot \frac{1}{\xi}, \quad (2.110)$$

where

$$\xi = (M \|\mathbf{d}^a\|^2 - \frac{M}{N} \delta_a^2)(N \|\mathbf{d}^b\|^2 - \frac{N}{M} \delta_b^2) - (\delta_a \delta_b - \rho)^2. \quad (2.111)$$

From (2.110) the bounds on the variances of the spatial and temporal frequency estimates are obtained as

$$CRB(\psi) = \frac{\gamma^2}{|\beta|^2} \cdot \frac{5N(\|\mathbf{d}^b\|^2 - \delta_b^2/M)}{3\xi} \quad (2.112)$$

and

$$CRB(\omega) = \frac{\gamma^2}{|\beta|^2} \cdot \frac{5M(\|\mathbf{d}^a\|^2 - \delta_a^2/N)}{3\xi}. \quad (2.113)$$

Then, by the invariance principle for maximum likelihood, we can find the bounds for ϕ and f . If we let $\mathbf{w} = \mathbf{W}(\Theta)$, we can express the Cramér-Rao bound for the estimator $\hat{\mathbf{w}}$ by means of the Cramér-Rao bound for $\hat{\Theta}$ [28]. In fact, we have the follow property for the two Fisher information matrices:

$$\mathbf{J}(\mathbf{w}) = \mathbf{GJ}(\Theta)\mathbf{G}^T, \quad (2.114)$$

where $\mathbf{G} = \{\partial\theta_j/\partial w_i\}$. In our case \mathbf{G} is diagonal

$$\mathbf{G} = \begin{bmatrix} \frac{2\pi d}{\lambda} \cos \phi & 0 & 0 & 0 \\ 0 & 2\pi T_r & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (2.115)$$

Using (2.106), (2.114) and again the partitioned inverse matrix lemma we obtain the bounds in (2.64) and (2.65). ■

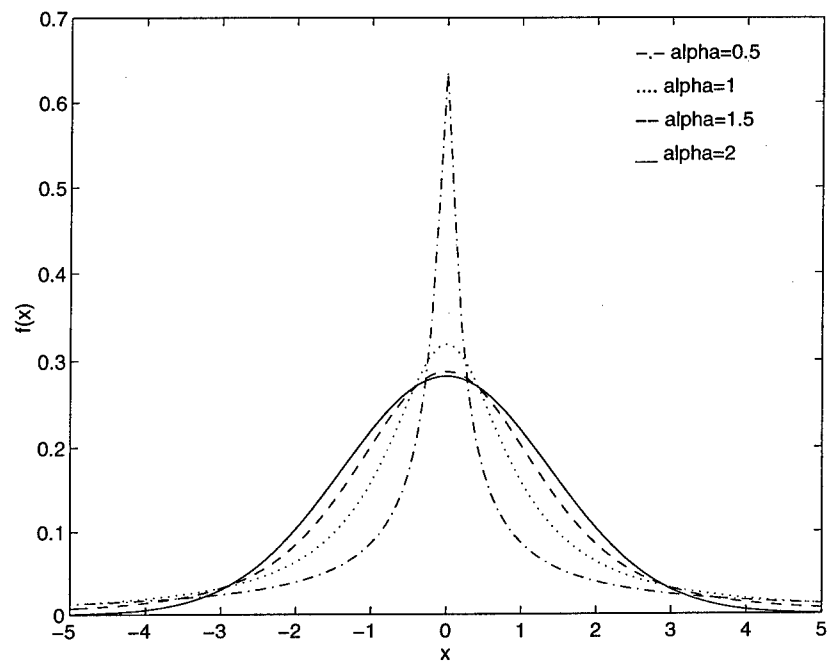


Figure 2.1: Standard $S\alpha S$ densities.

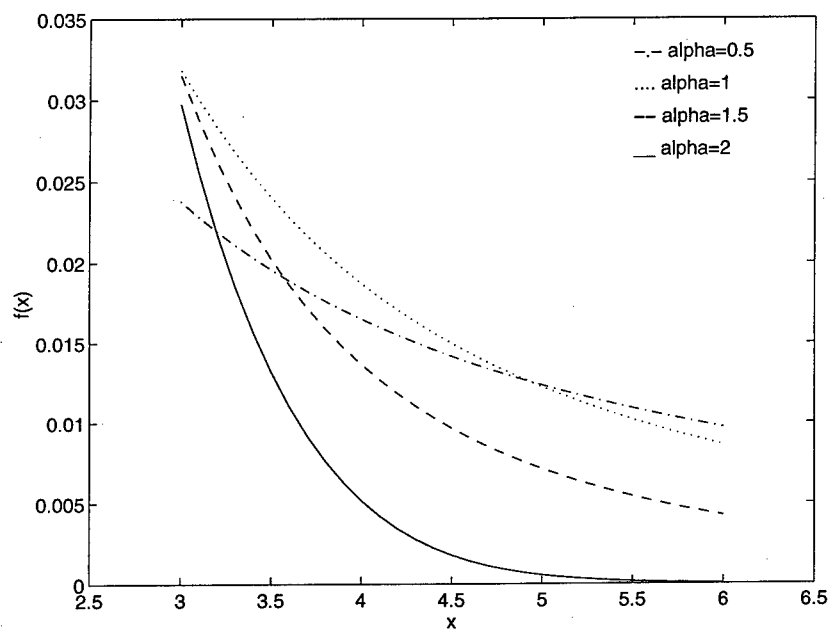


Figure 2.2: A close-up view of the tails of the densities in Figure 2.1.

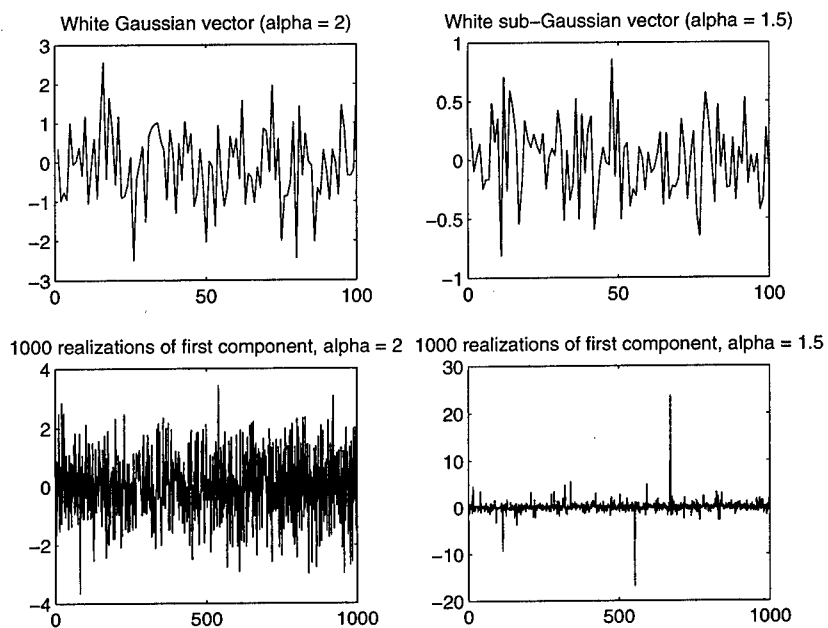


Figure 2.3: Typical realizations of sub-Gaussian random vectors.

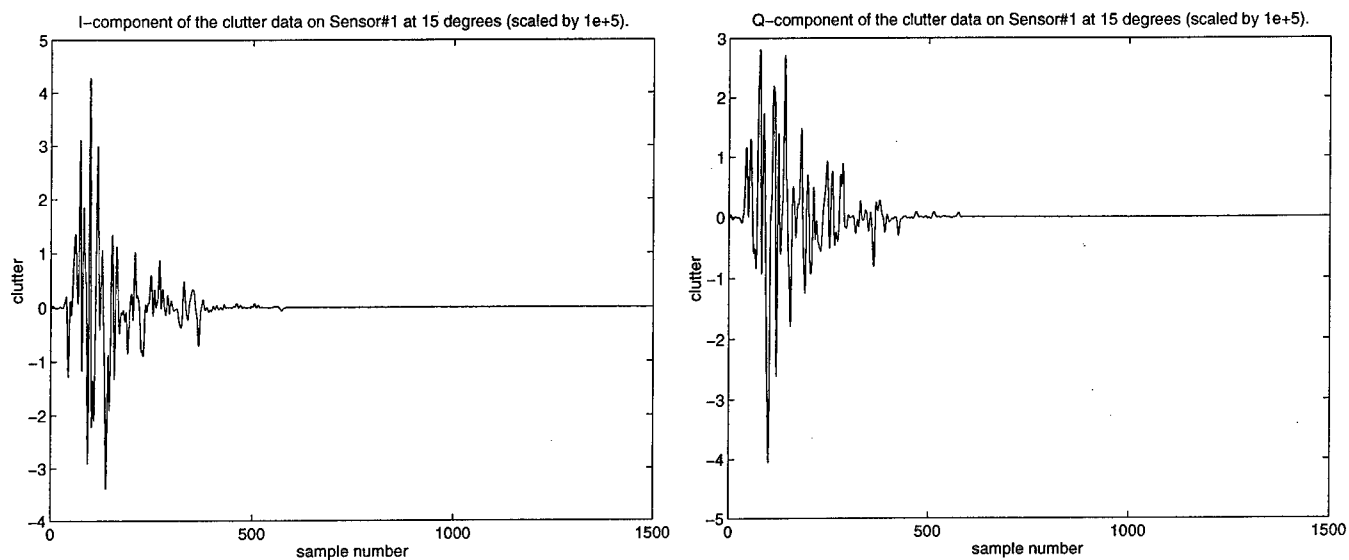


Figure 2.4: Measured I/Q-components of radar clutter: Azimuth: 15° , estimated mean: $0.0043 + j0.0099$, estimated $\alpha = 0.7318$, $\gamma = 0.0413$.

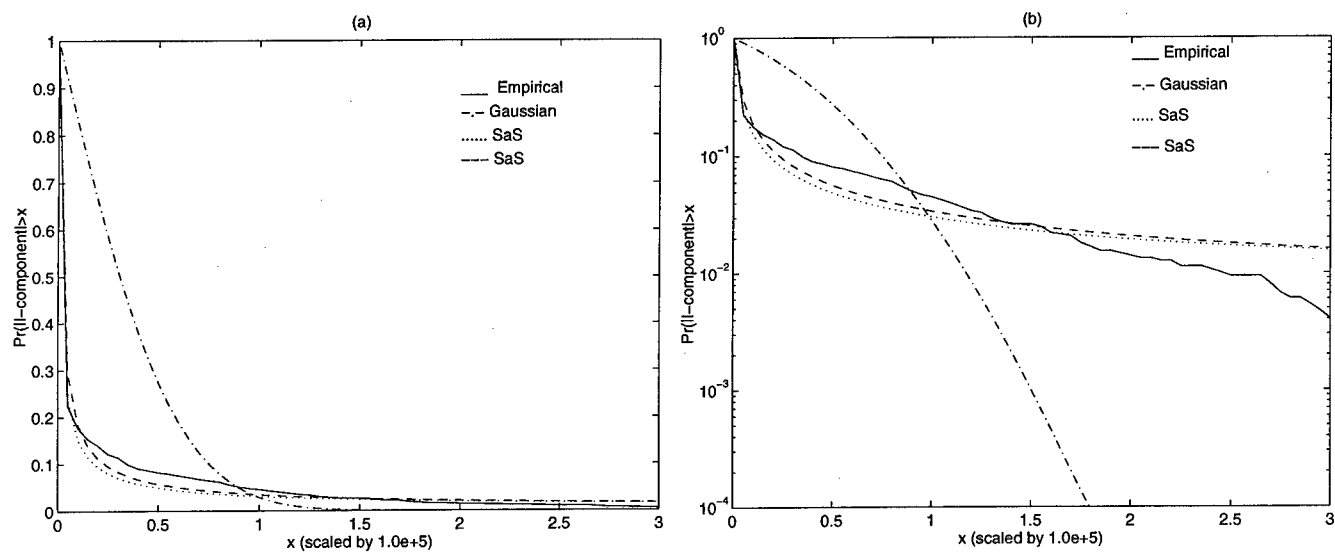


Figure 2.5: Comparison of the empirical, SaS, and Gaussian amplitude probability distributions.

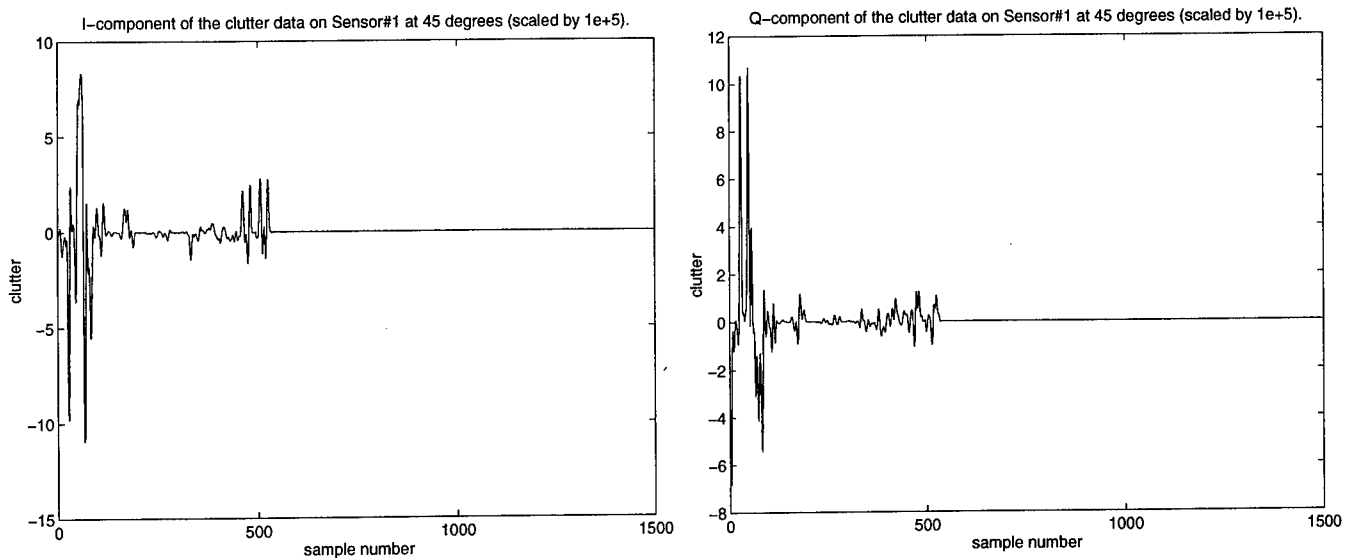


Figure 2.6: Measured I/Q-components of radar clutter: Azimuth: 45° , estimated mean: $-0.0121 - j0.0247$, estimated $\alpha = 0.8301$, $\gamma = 0.0537$.

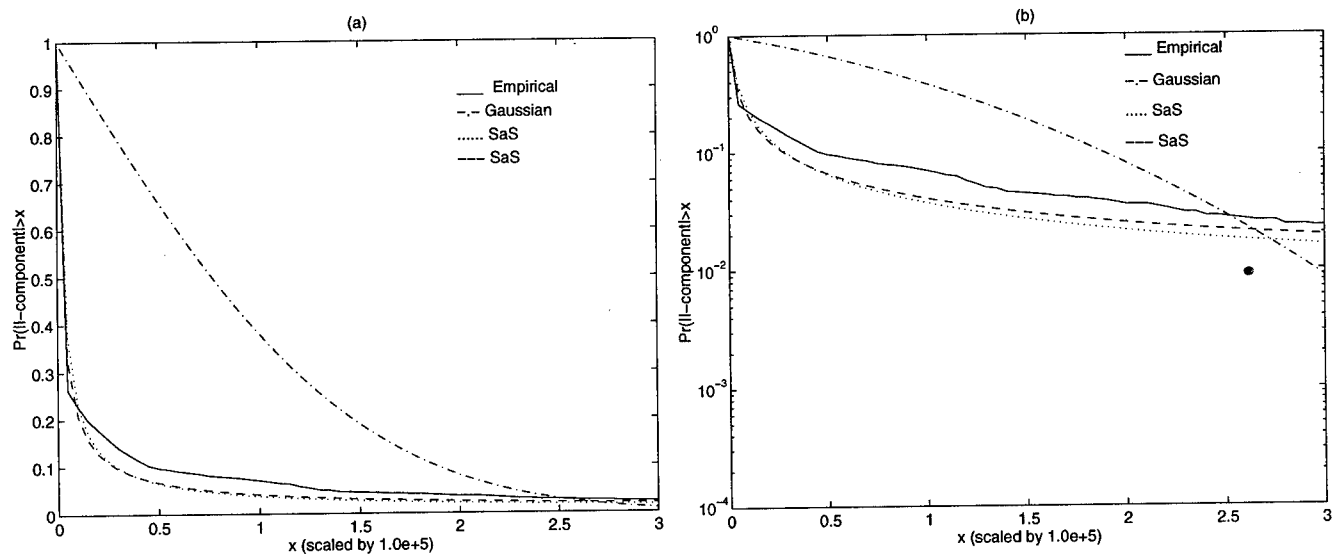


Figure 2.7: Comparison of the empirical, SaS, and Gaussian amplitude probability distributions.

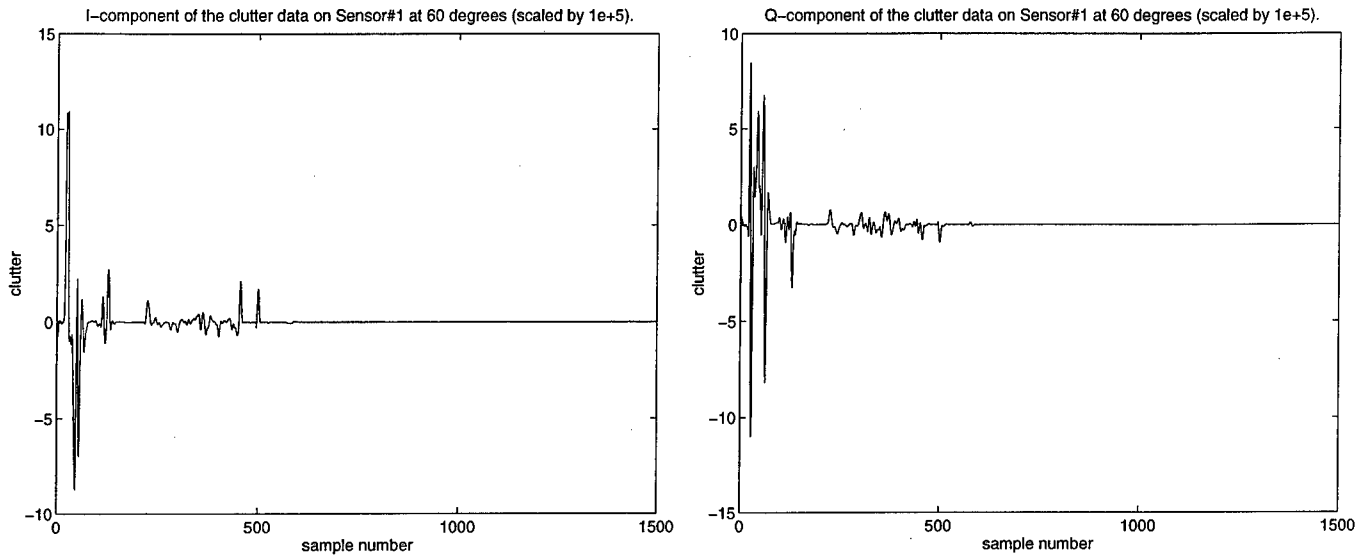


Figure 2.8: Measured I/Q-components of radar clutter: Azimuth: 60° , estimated mean: $0.0051 + j0.005$, estimated $\alpha = 0.6886$, $\gamma = 0.0394$.

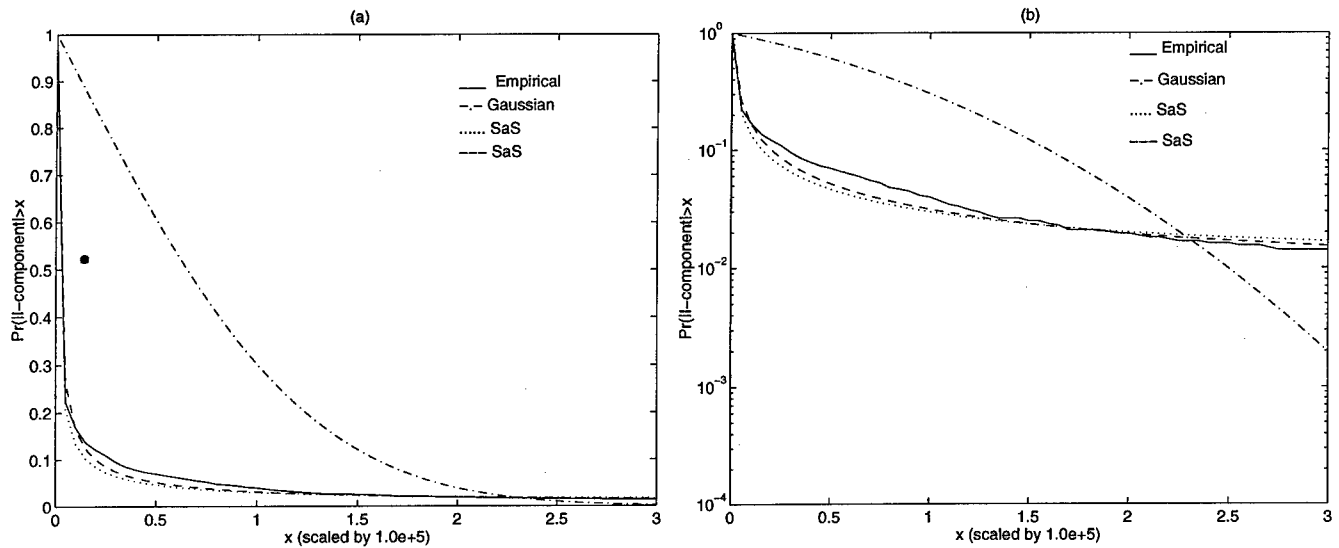


Figure 2.9: Comparison of the empirical APD with Gaussian and SaS models.

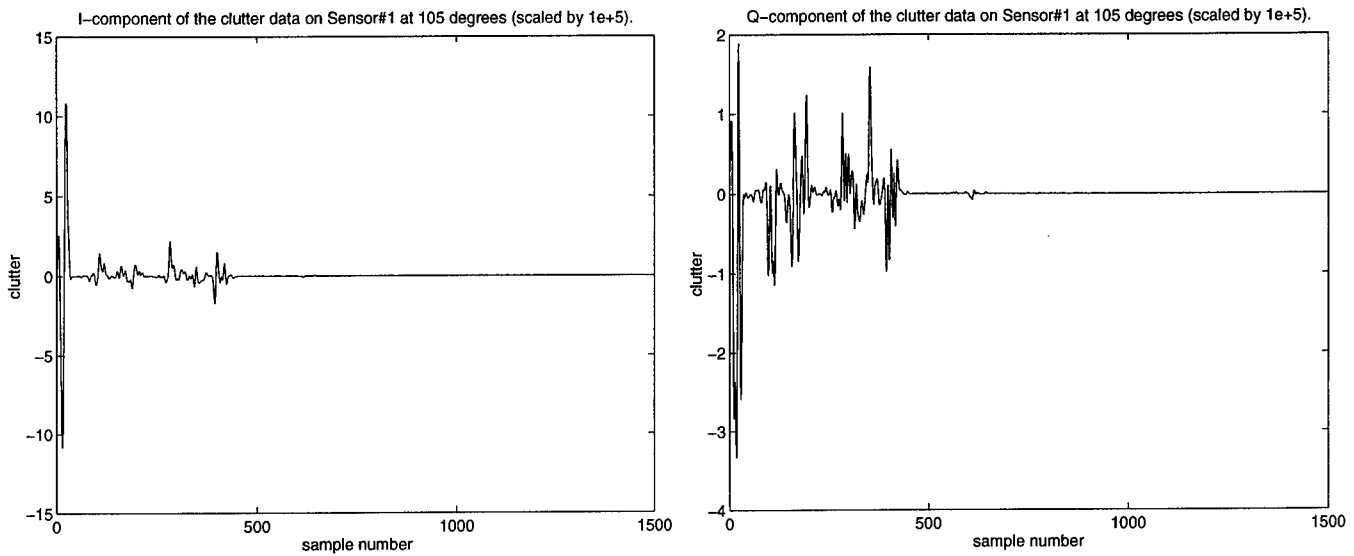


Figure 2.10: Measured I/Q-components of radar clutter: Azimuth: 105° , estimated mean: $0.028 - j0.0241$, estimated $\alpha = 1.1045$, $\gamma = 0.0249$.

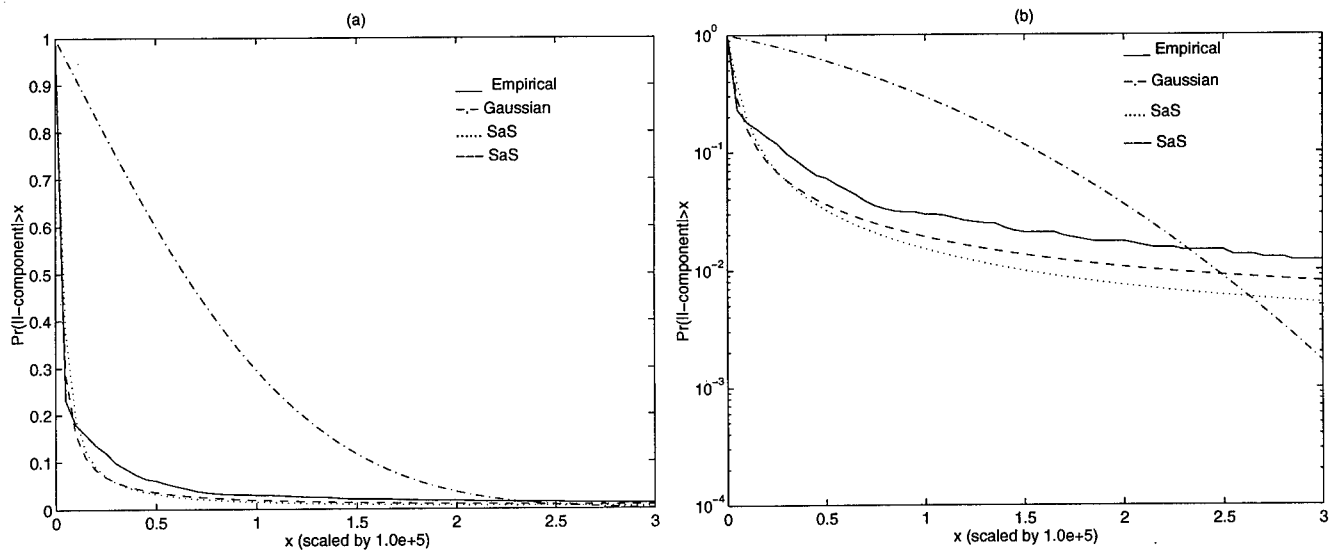


Figure 2.11: Comparison of the empirical APD with Gaussian and SaS models.

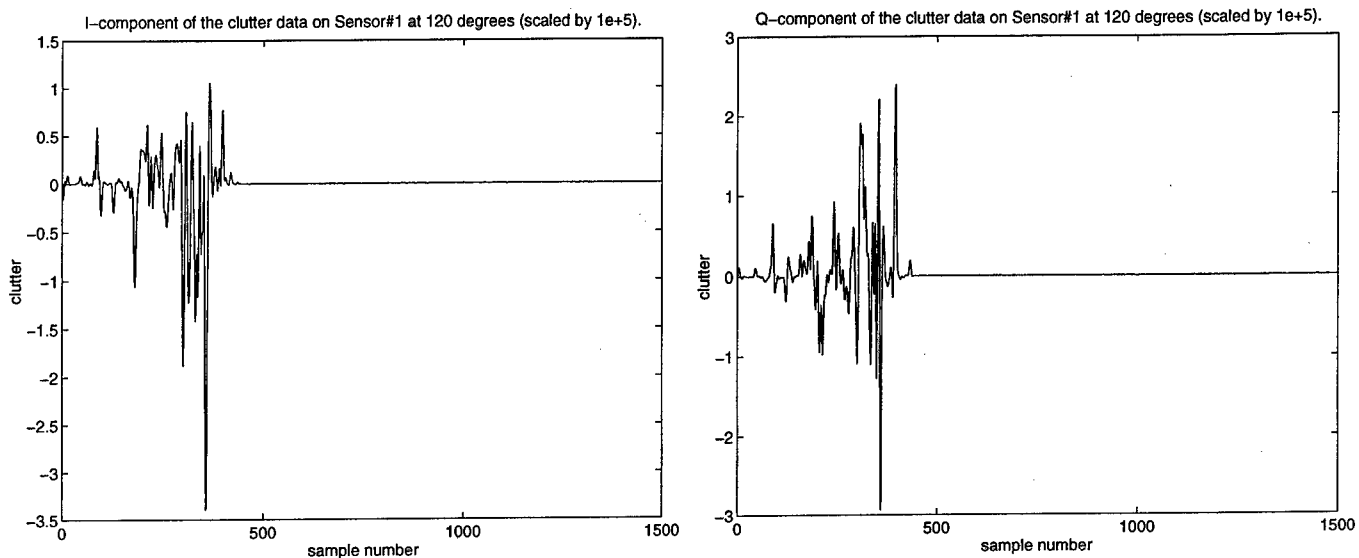


Figure 2.12: Measured I/Q-components of radar clutter: Azimuth: 120° , estimated mean: $-0.0199 + j0.0215$, estimated $\alpha = 1.1314$, $\gamma = 0.0171$.

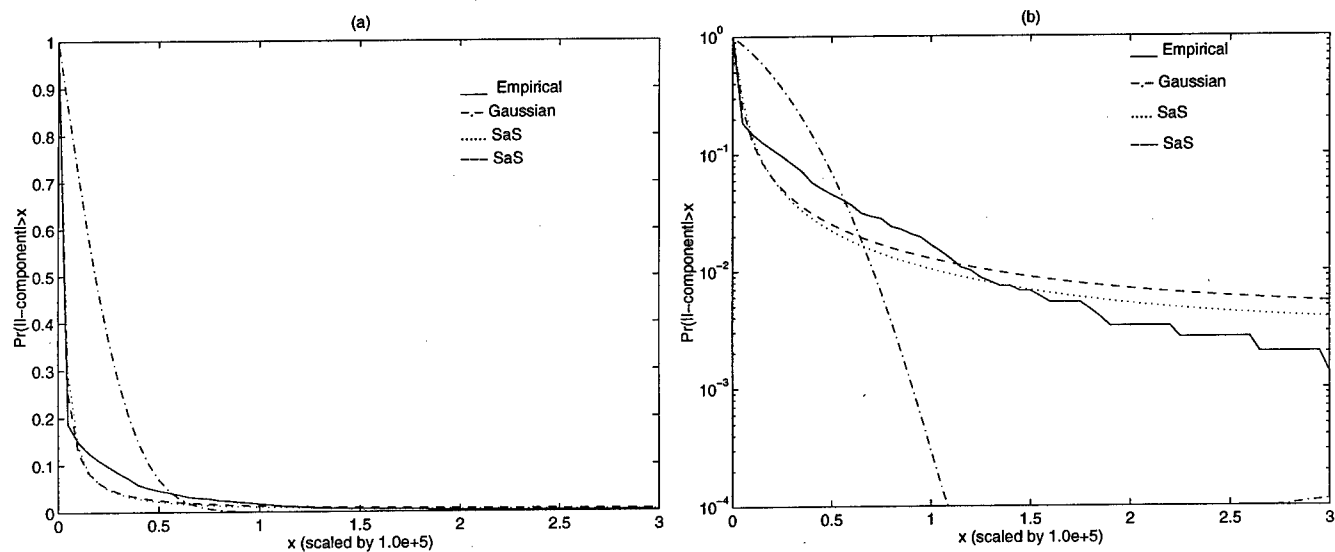


Figure 2.13: Comparison of the empirical APD with Gaussian and SaS models.

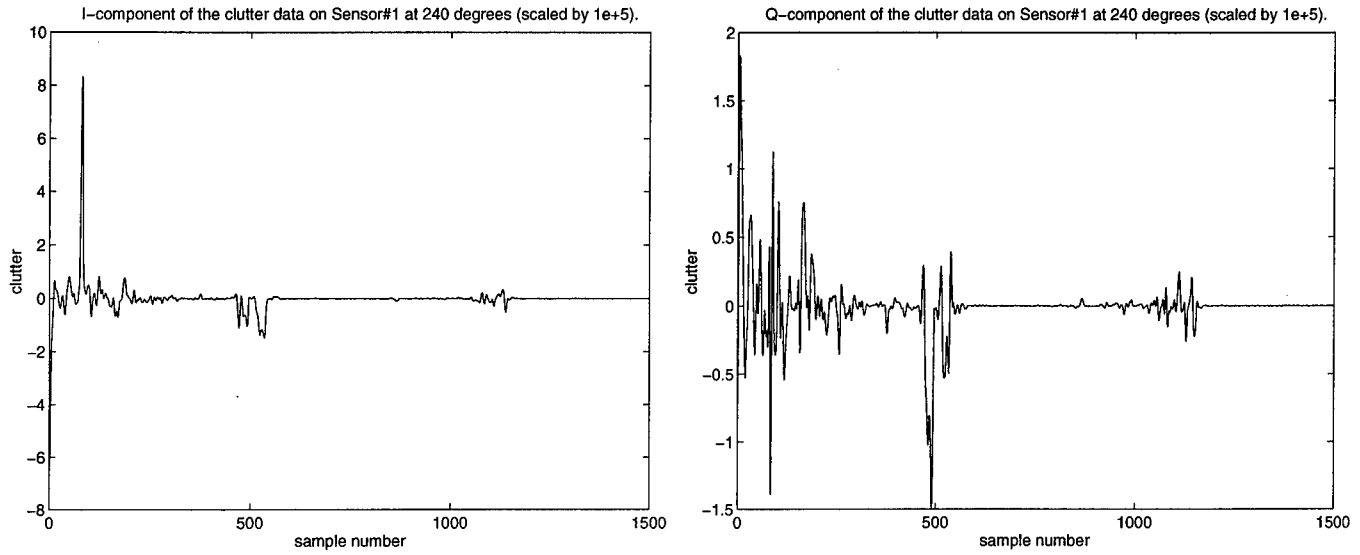


Figure 2.14: Measured I/Q-components of radar clutter: Azimuth: 240° , estimated mean: $-0.0092 - j0.007$, estimated $\alpha = 0.8358$, $\gamma = 0.0337$.

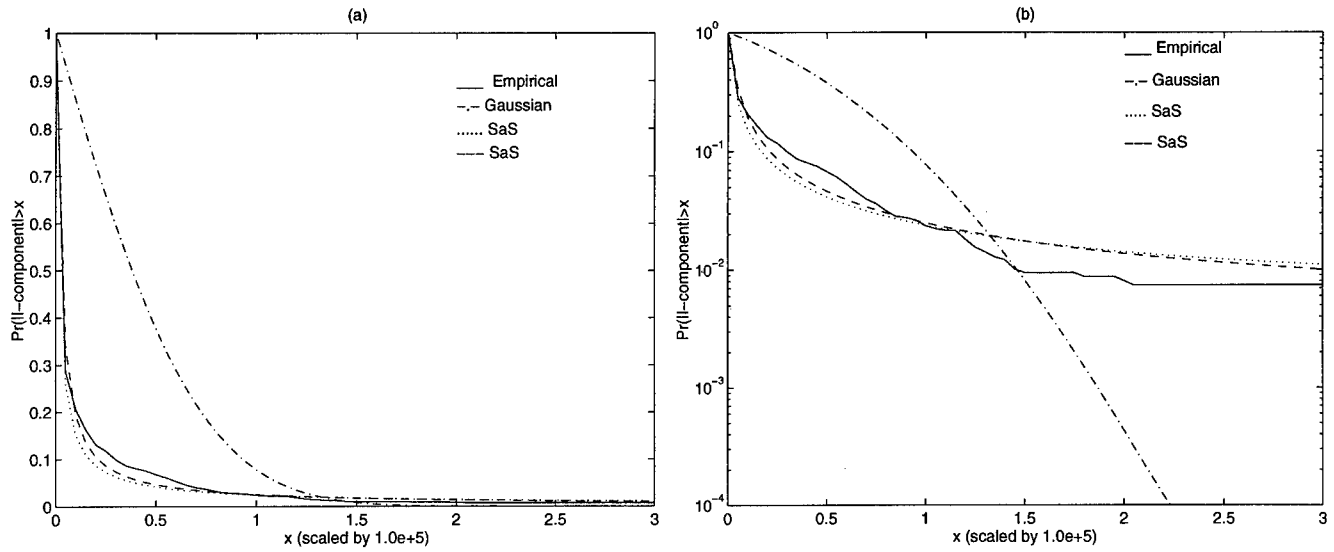


Figure 2.15: Comparison of the empirical APD with Gaussian and SaS models.

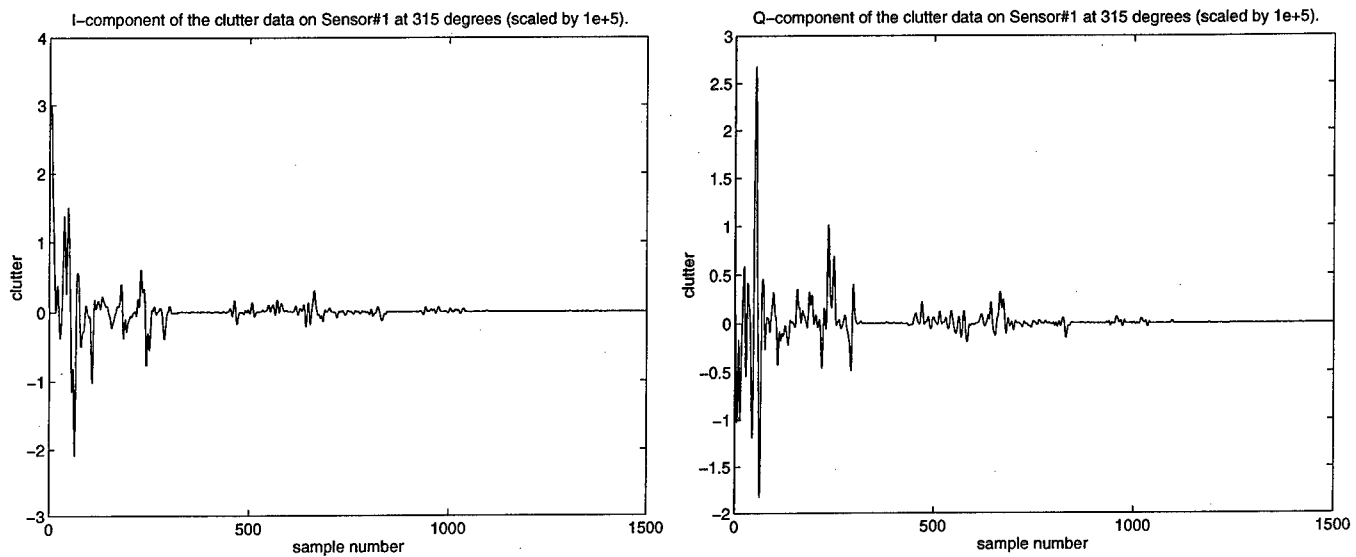


Figure 2.16: Measured I/Q-components of radar clutter: Azimuth: 315° , estimated mean: $0.017 + j0.0079$, estimated $\alpha = 1.0304$, $\gamma = 0.0193$.

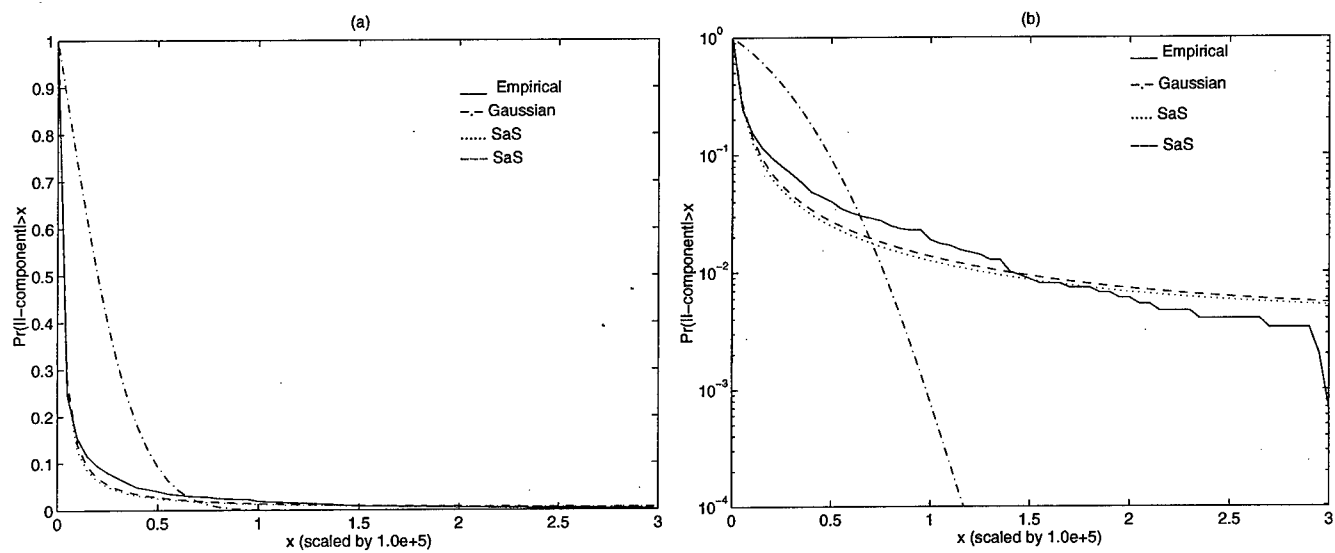


Figure 2.17: Comparison of the empirical APD with Gaussian and SaS models.

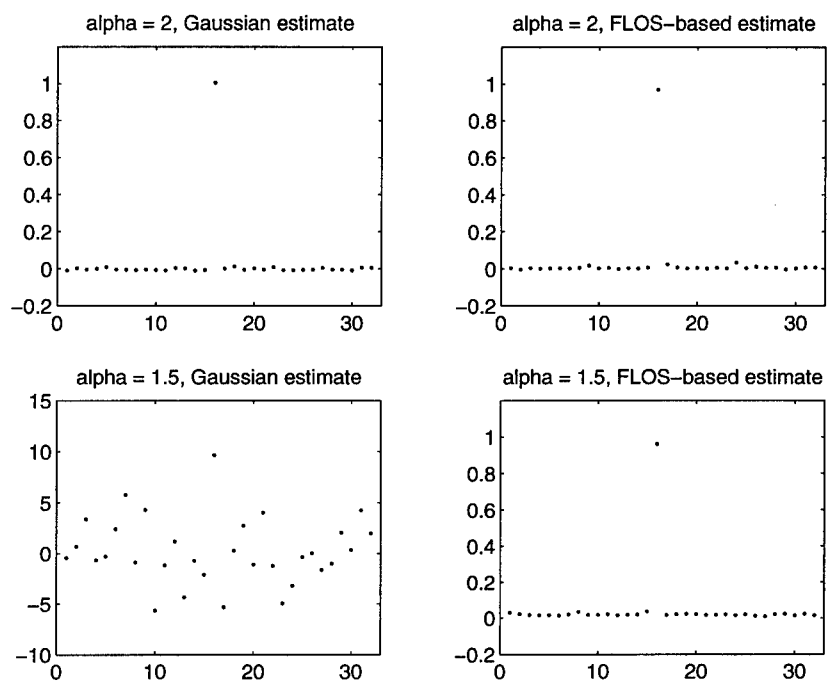


Figure 2.18: Illustration of the performance of estimators of the underlying matrix of a subGaussian vector.

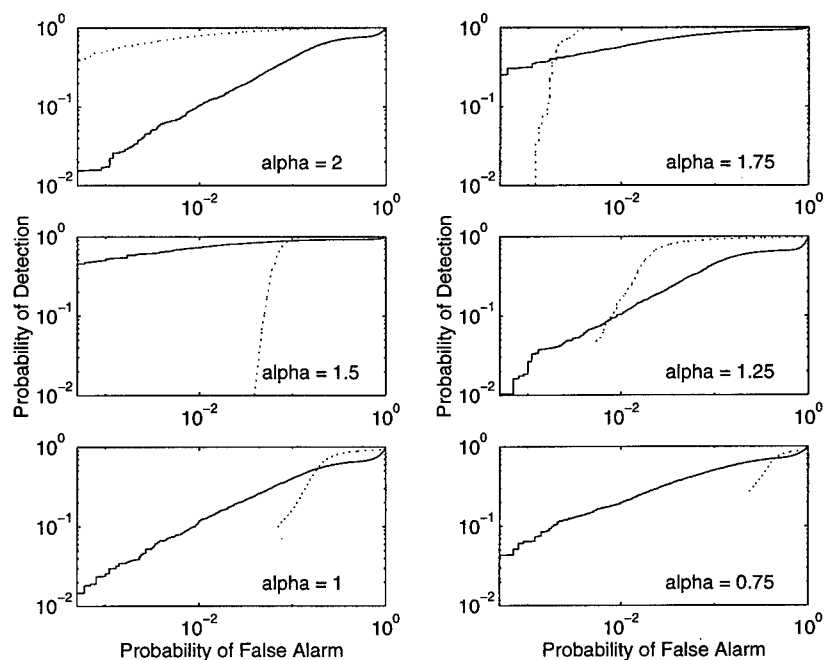


Figure 2.19: Comparison of the small sample performance of the Gaussian (dotted line) and the Cauchy (solid line) detector.

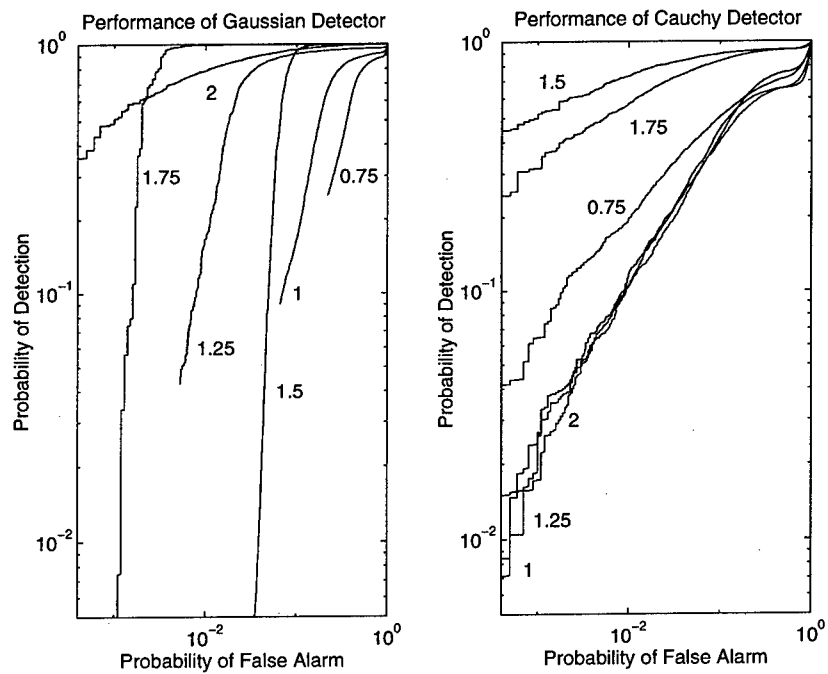


Figure 2.20: Performance of the Gaussian (left column) and the Cauchy (right column) detector as a function of the characteristic exponent α .

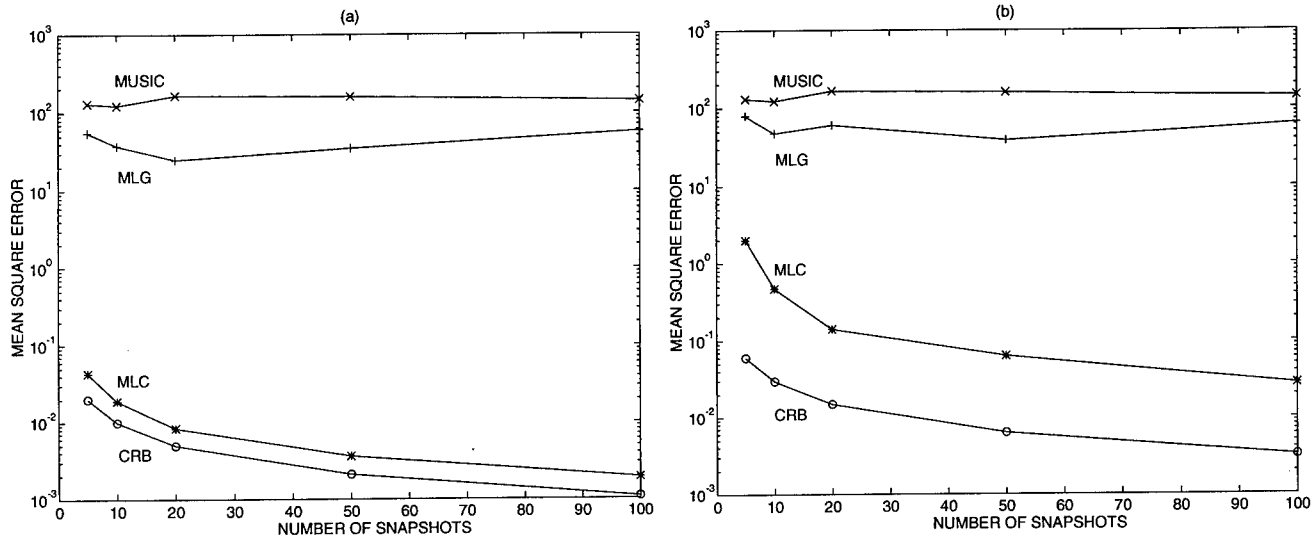


Figure 2.21: MSE of the estimated DOA and CRB as functions of the number of snapshots M . (a) Exact signal knowledge, (b) Least-squares estimate of the signal.

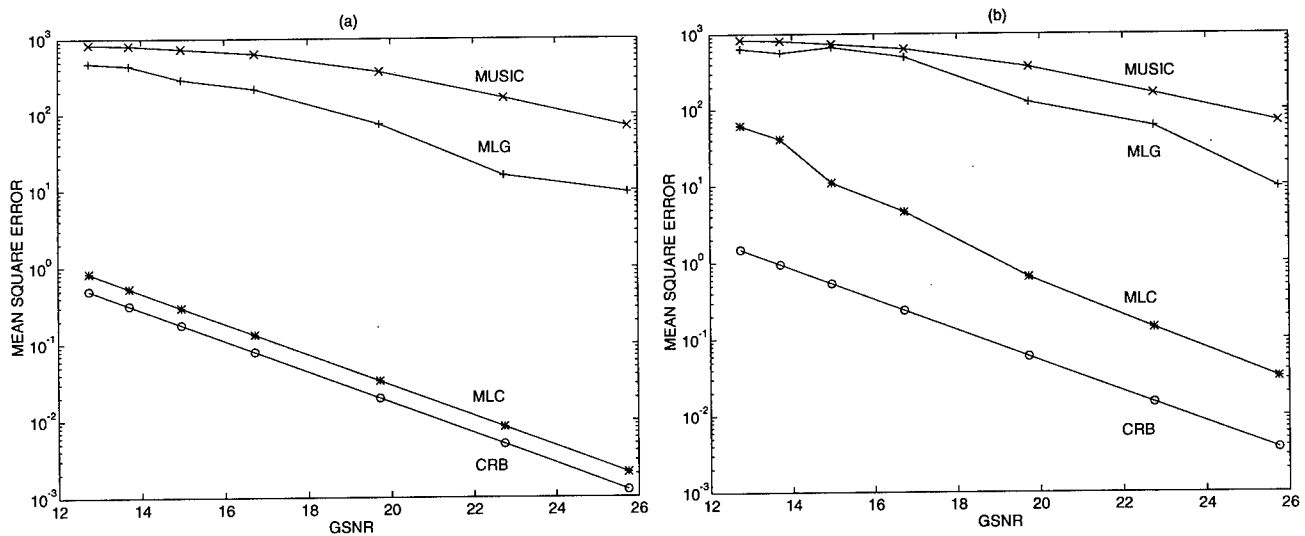


Figure 2.22: MSE of the estimated DOA and CRB as functions of the GSNR. (a) Exact signal knowledge, (b) Least-squares estimate of the signal.

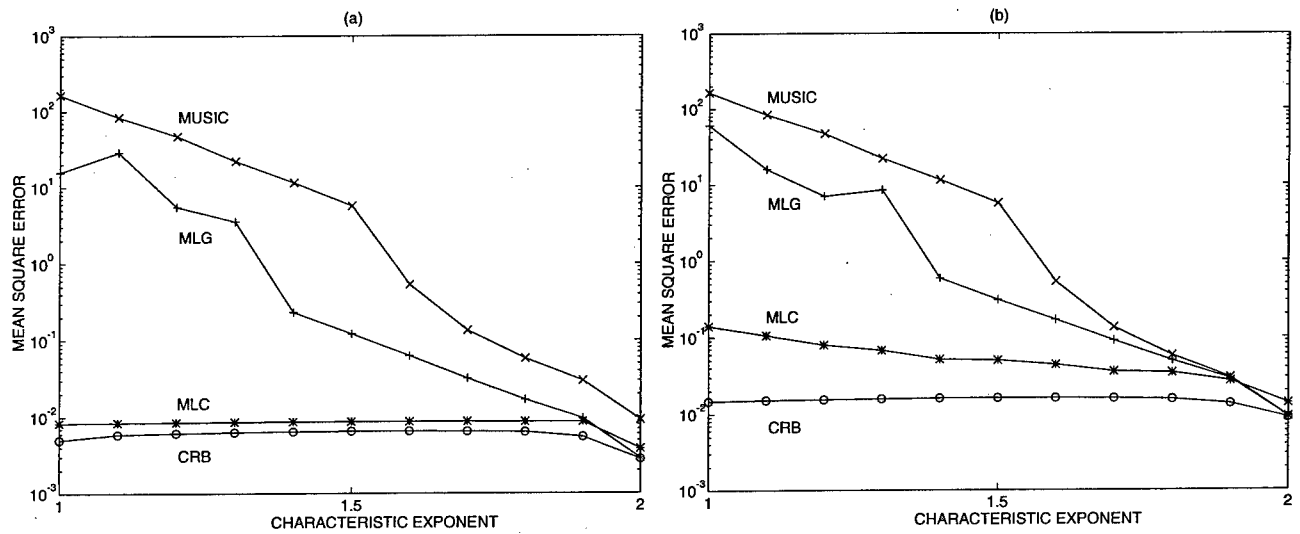


Figure 2.23: MSE of the estimated DOA and CRB as functions of the characteristic exponent α .
 (a) Exact signal knowledge, (b) Least-squares estimate of the signal.

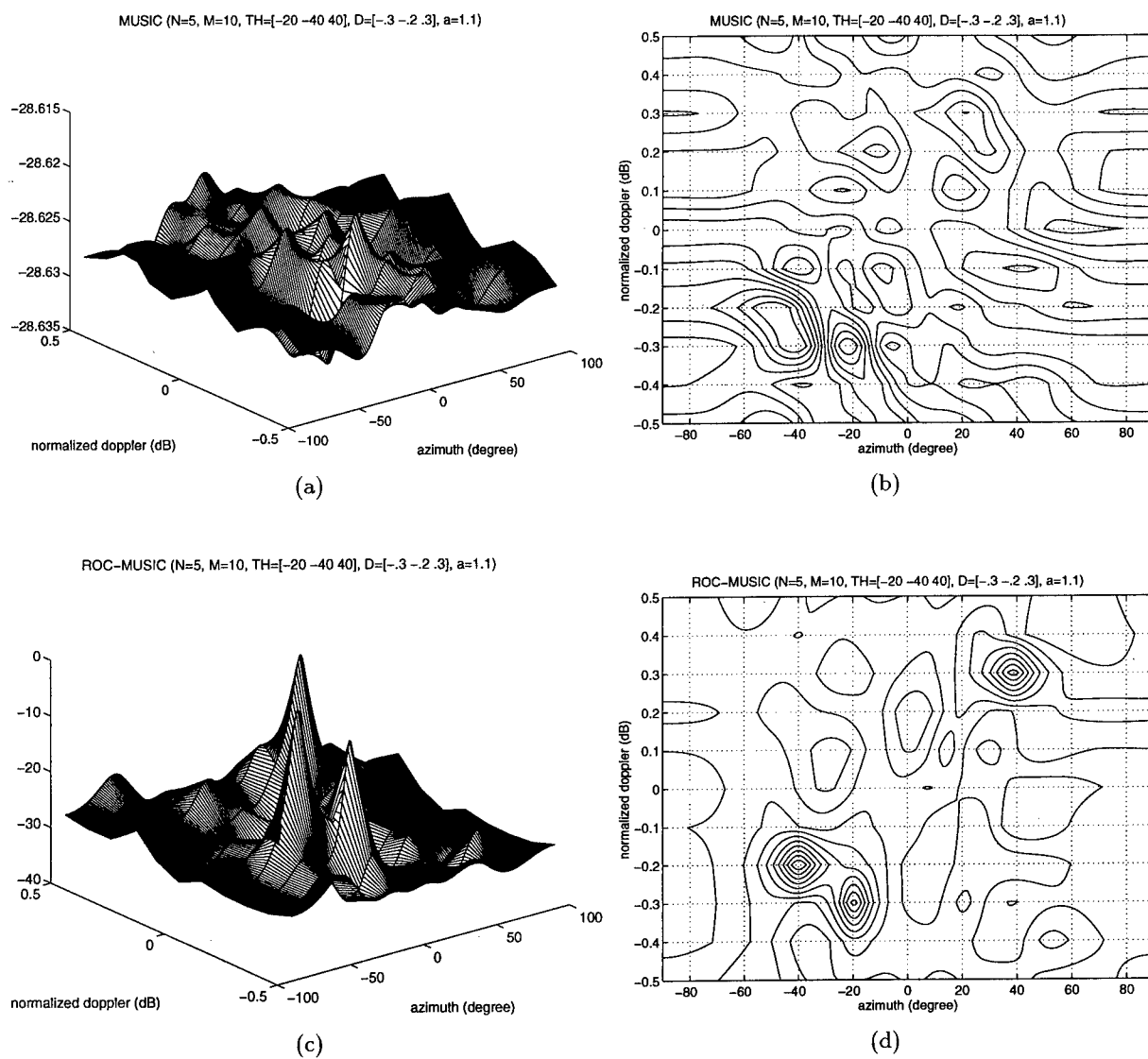


Figure 2.24: MUSIC (a-b) and ROC-MUSIC (c-d) angle-Doppler spectra ($N = 5$, $M = 10$, $\Theta = [-20^\circ, -40^\circ, 40^\circ]$ $D = [-0.3, -0.2, 0.3]$). Additive stable noise ($\alpha = 1.1$, $\gamma = 2$).

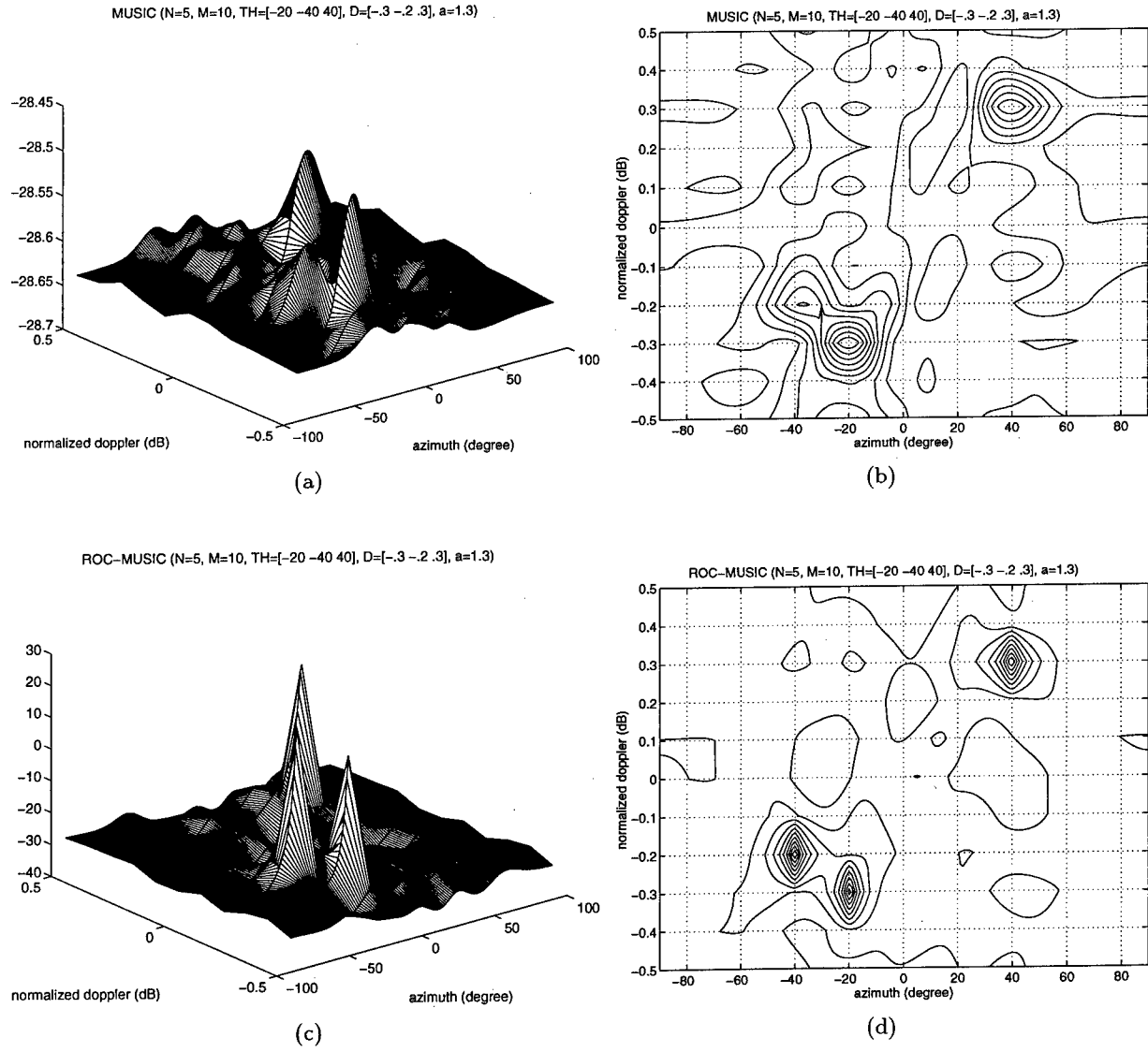


Figure 2.25: MUSIC (a-b) and ROC-MUSIC (c-d) angle-Doppler spectra ($N = 5$, $M = 10$, $\Theta = [-20^\circ, 30^\circ, 40^\circ]$ $\mathbf{D} = [-0.3, 0.2, 0.3]$). Additive stable noise ($\alpha = 1.3$, $\gamma = 2$).

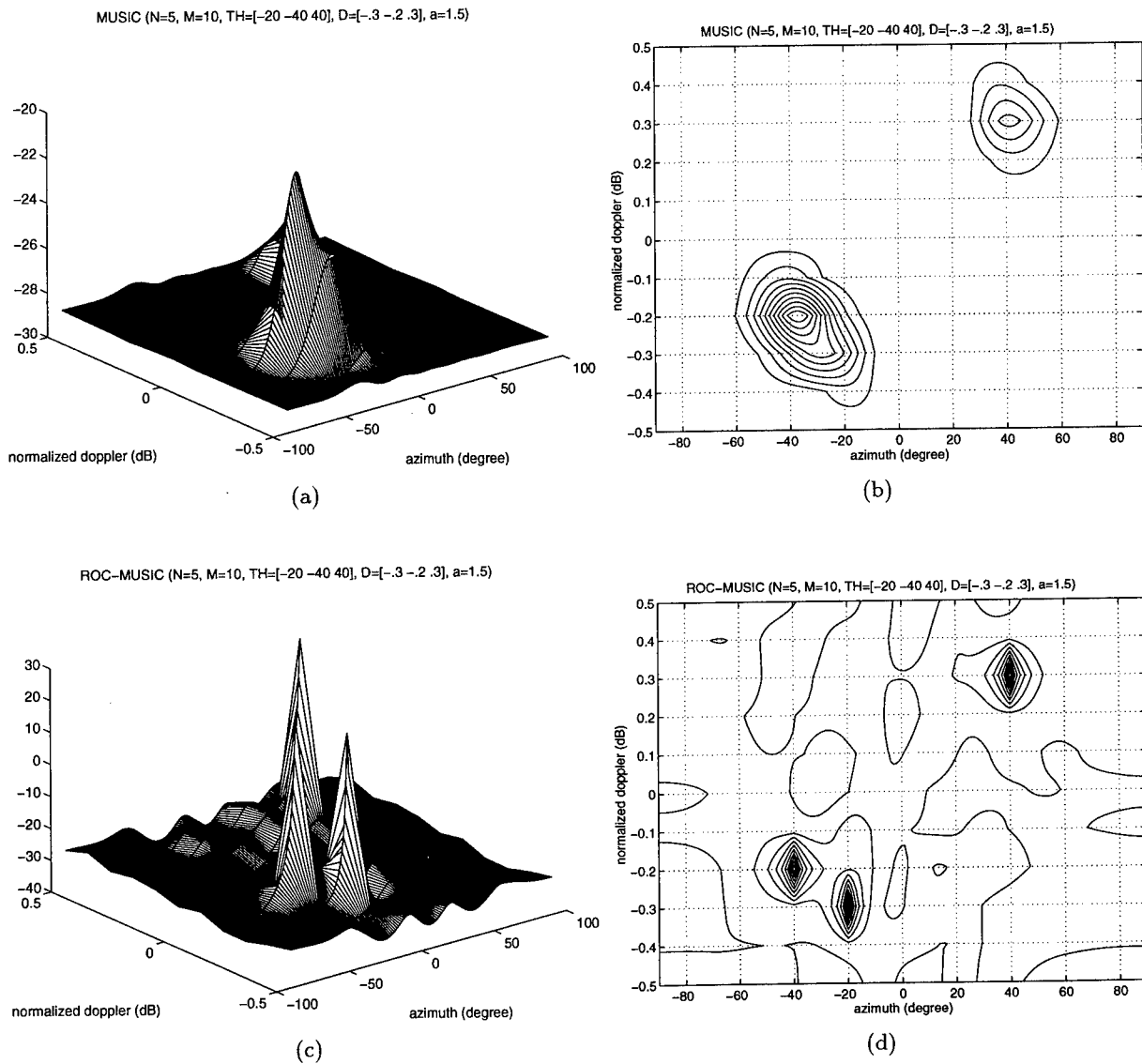


Figure 2.26: MUSIC (a-b) and ROC-MUSIC (c-d) angle-Doppler spectra ($N = 5$, $M = 10$, $\Theta = [-20^\circ, -40^\circ, 40^\circ]$ $\mathbf{D} = [-0.3, -0.2, 0.3]$). Additive stable noise ($\alpha = 1.5$, $\gamma = 4$).

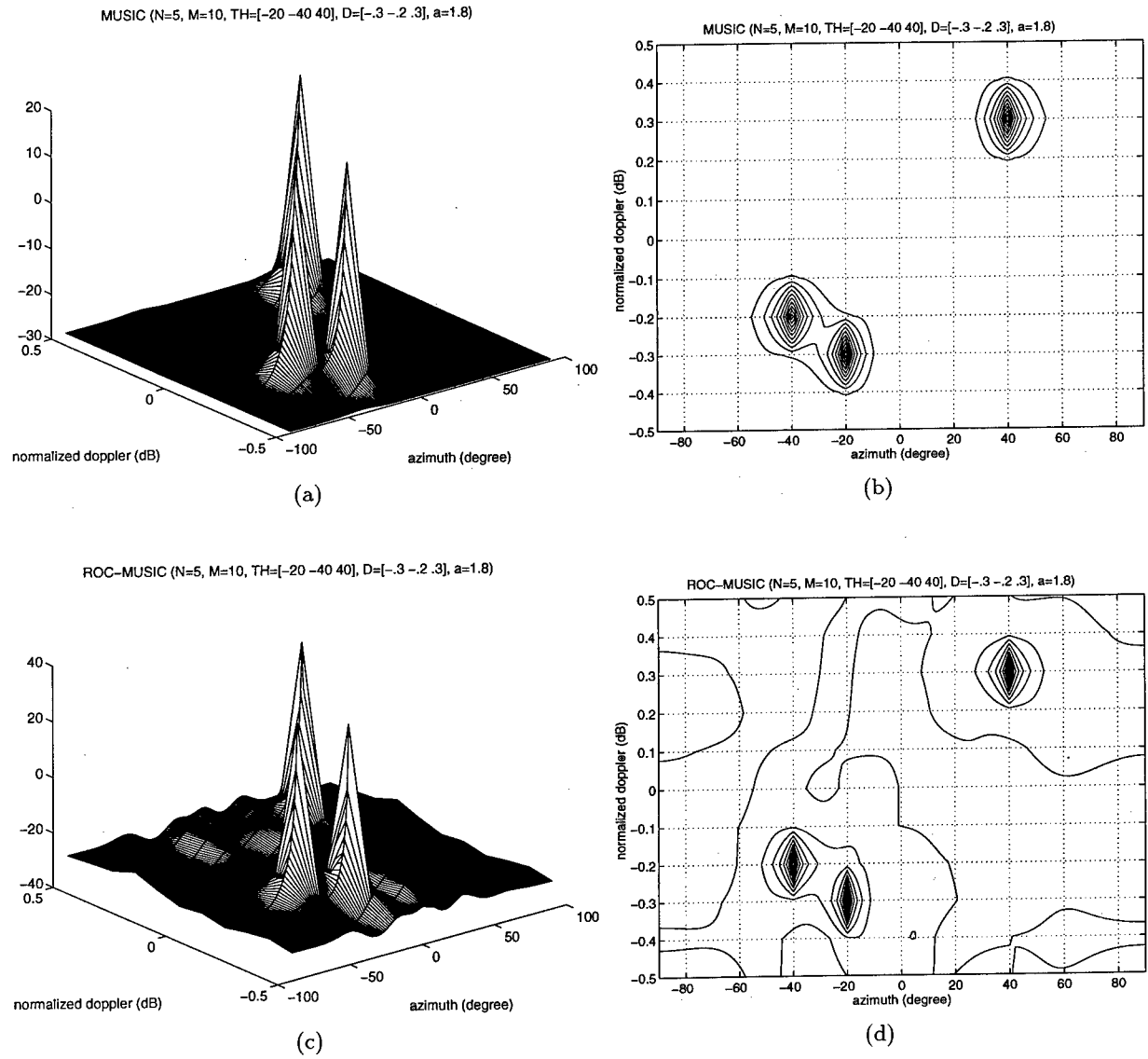


Figure 2.27: MUSIC (a-b) and ROC-MUSIC (c-d) angle-Doppler spectra ($N = 5$, $M = 10$, $\Theta = [-20^\circ, -40^\circ, 40^\circ]$ $D = [-0.3, -0.2, 0.3]$). Additive stable noise ($\alpha = 1.8$, $\gamma = 4$).

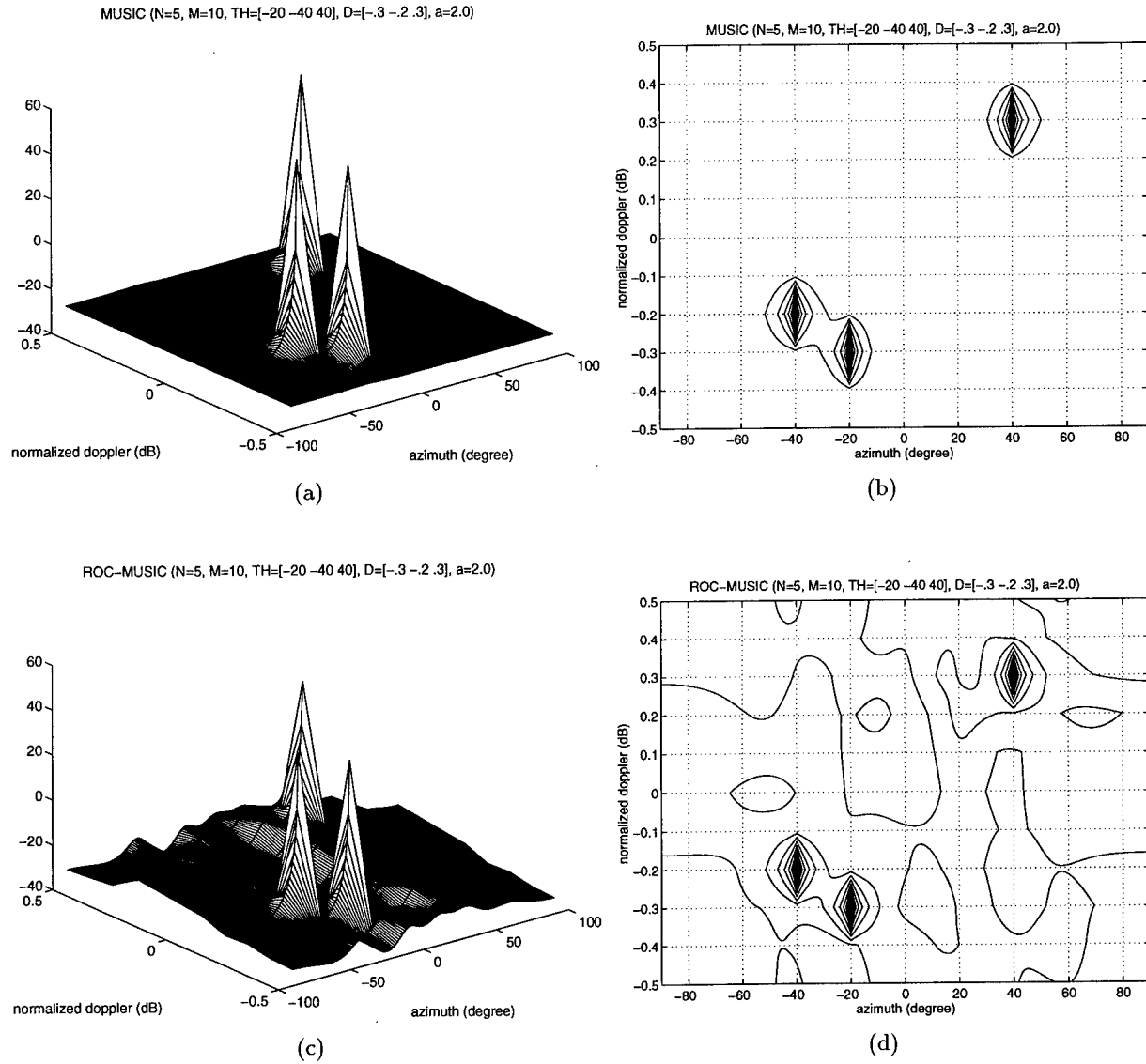


Figure 2.28: MUSIC (a-b) and ROC-MUSIC (c-d) angle-Doppler spectra ($N = 5$, $M = 10$, $\Theta = [-20^\circ, -40^\circ, 40^\circ]$ $\mathbf{D} = [-0.3, -0.2, 0.3]$). Additive Gaussian noise ($\alpha = 2.0$, $\gamma = 4$).

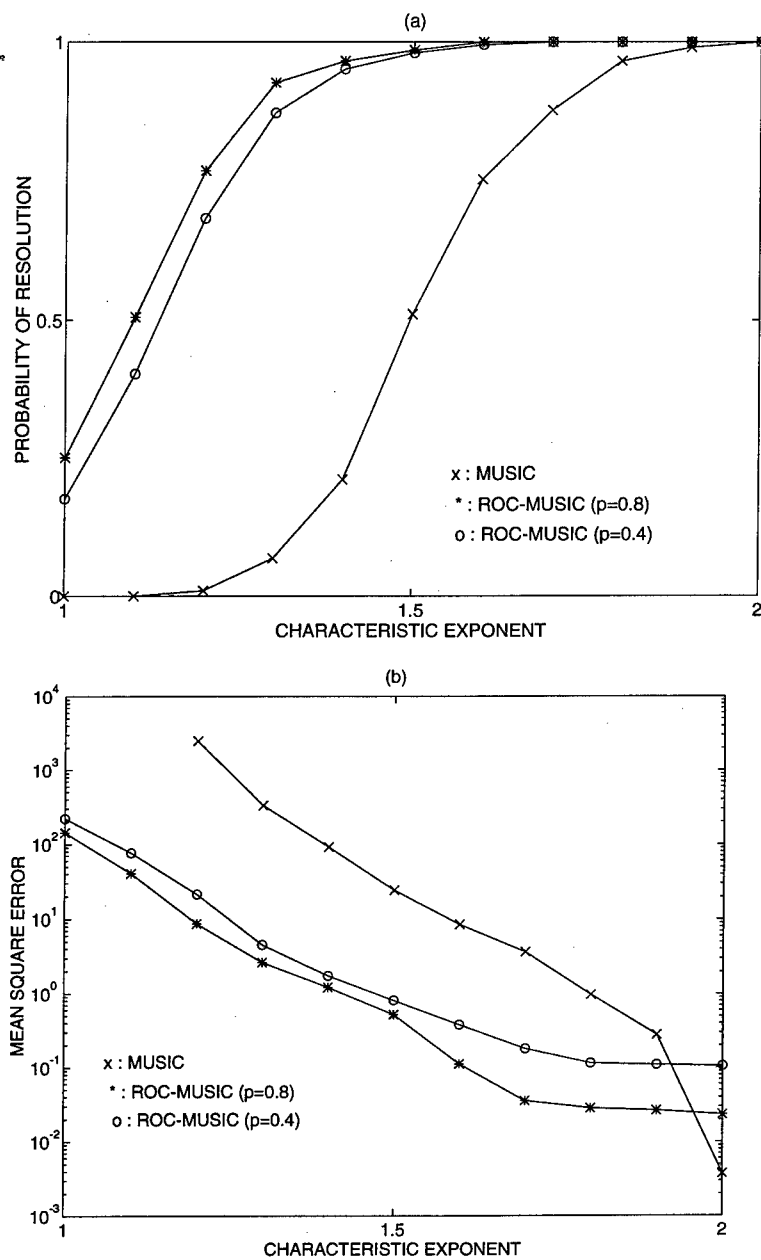


Figure 2.29: Probability of resolution (a) and mean square error (b) as functions of the characteristic exponent α .

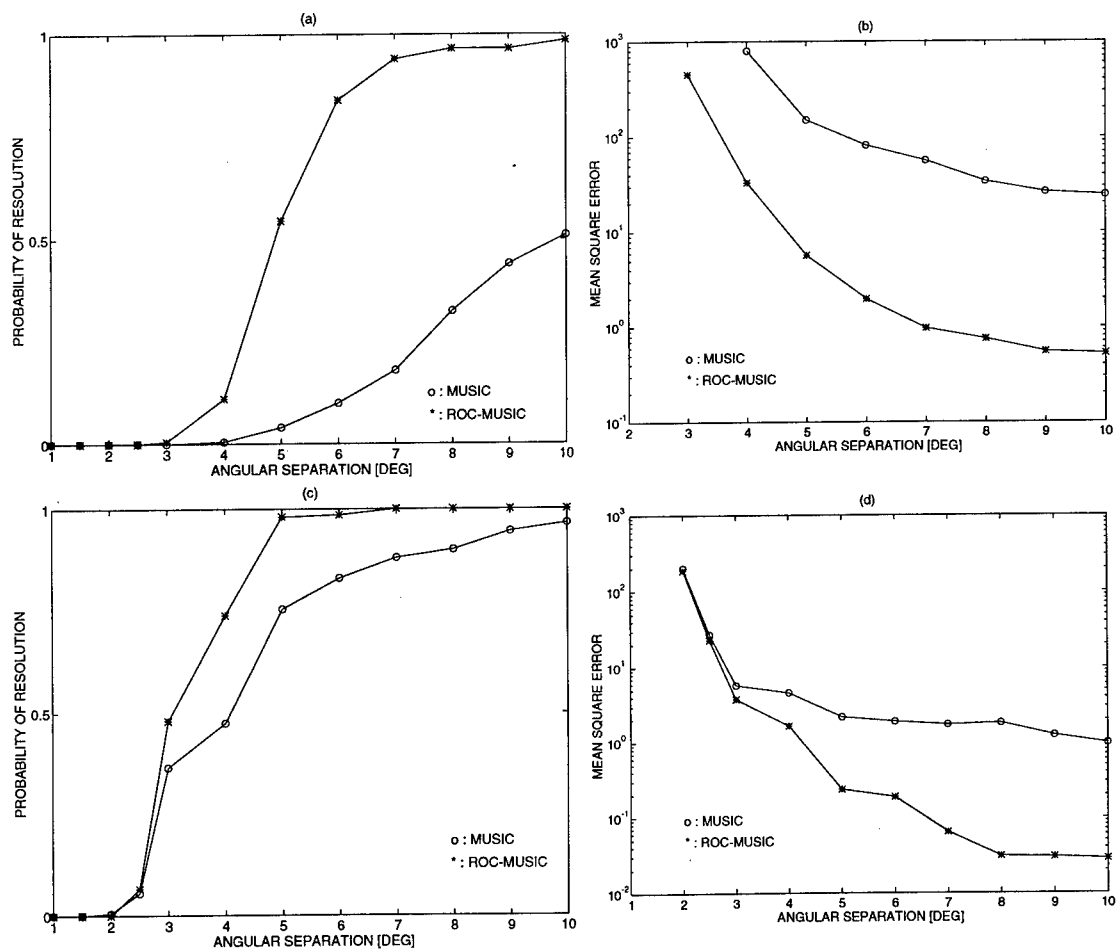


Figure 2.30: Probability of resolution and mean square error as functions of the source angular separation, (a-b): $\alpha = 1.5$, (c-d): $\alpha = 1.8$.

Bibliography

- [1] C. L. Nikias and M. Shao, *Signal Processing with Alpha-Stable Distributions and Applications*. New York: John Wiley and Sons, 1995.
- [2] S. Cambanis, G. Samorodnitsky, and M. S. Taqqu, eds., *Stable Processes and Related Topics*. Boston: Birkhauser, 1991.
- [3] S. Cambanis, "Complex symmetric stable variables and processes," in *Contributions to Statistics: Essays in Honor of Norman L. Johnson* (P. Sen, ed.), pp. 63–79, New York: North-Holland, 1983.
- [4] S. Cambanis, C. D. Hardin, and A. Weron, "Innovations and Wold decompositions of stable sequences," *Probab. Th. Rel. Fields*, vol. 79, pp. 1–27, 1988.
- [5] V. M. Zolotarev, "Integral transformations of distributions and estimates of parameters of multidimensional spherically symmetric stable laws," in *Contributions to Probability* (J. Gani and V. K. Rohatgi, eds.), pp. 283–305, New York: Academic Press, 1981.
- [6] S. Cambanis, C. D. Hardin, and A. Weron, "Ergodic properties of stationary stable processes," *Stochastic Processes and their Applications*, vol. 24, pp. 1–18, 1987.
- [7] E. Masry and S. Cambanis, "Spectral density estimation for stationary stable processes," *Stochastic Processes and their Applications*, vol. 18, pp. 1–31, 1984.
- [8] S. Cambanis and A. G. Miamee, "On prediction of harmonizable stable processes," *Sankhyā: The Indian Journal of Statistics*, vol. 51, pp. 269–294, 1989.
- [9] W. Wu and S. Cambanis, "Conditional variance of symmetric stable variables," in *Stable Processes and Related Topics* (S. Cambanis, G. Samorodnitsky, and M. S. Taqqu, eds.), pp. 85–99, Boston: Birkhauser, 1991.
- [10] G. Samorodnitsky and M. S. Taqqu, *Stable Non-Gaussian Random Processes: Stochastic Models with Infinite Variance*. New York: Chapman and Hall, 1994.
- [11] A. Janicki and A. Weron, *Simulation and Chaotic Behavior of α -Stable Stochastic Processes*. New York: Dekker, 1993.
- [12] M. Shao and C. L. Nikias, "Signal processing with fractional lower order moments: Stable processes and their applications," *Proc. IEEE*, vol. 81, pp. 986–1010, 1993.
- [13] S. Cambanis and G. Miller, "Linear problems in p th order and stable processes," *SIAM J. Appl. Math.*, vol. 41, pp. 43–69, 1981.

- [14] G. Samorodnitsky and M. S. Taquq, *Stable, Non-Gaussian Random Processes: Stochastic Models with Infinite Variance*. New York, NY: Chapman & Hall, 1994.
- [15] S. Cambanis and A. R. Soltani, "Prediction of stable processes: Spectral and moving average representations," *Z. Wahrsch verw. Gebiete*, vol. 66, pp. 593–612, 1984.
- [16] J. M. Chambers, C. L. Mallows, and B. W. Stuck, "A method of simulating stable random variables," *Journal of the American Statistical Association*, vol. 71, pp. 340–344, 1976.
- [17] X. Ma and C. L. Nikias, "Parameter estimation and blind channel identification for impulsive signal environments," *IEEE Trans. Signal Processing*, vol. 43, pp. 2884–2897, December 1995.
- [18] M. Abramowitz and I. A. Stegun, eds., *Handbook of Mathematical Functions*. New York: Dover Publications, 1965.
- [19] G. A. Tsihrintzis and C. L. Nikias, "Fast estimation of the parameters of alpha-stable impulsive interference," *IEEE Trans. Signal Processing*, June 1996.
- [20] H. Cramér, *Mathematical Methods of Statistics*. Princeton, NJ: Princeton University Press, 1946.
- [21] V. Zolotarev, *One-Dimensional Stable Distributions*. Providence, RI: American Mathematical Society, 1986.
- [22] P. J. Bickel and K. A. Doksum, *Mathematical Statistics: Basic Ideas and Selected Topics*. San Fransico, CA: Holden-Day, 1977.
- [23] J. Ward, "Space-time adaptive processing for airborne radar," Tech. Rep. 1015, Lincoln Laboratory, Dec. 1994.
- [24] P. Tsakalides and C. L. Nikias, "Maximum likelihood localization of sources in noise modeled as a stable process," *IEEE Trans. Signal Processing*, vol. 43, pp. 2700–2713, Nov. 1995.
- [25] D. G. Luenberger, *Linear and Nonlinear Programming*. Menlo Park: Addison Wesley, second ed., 1984.
- [26] P. Tsakalides and C. L. Nikias, "The robust covariation-based MUSIC (ROC-MUSIC) algorithm for bearing estimation in impulsive noise environments," *IEEE Trans. Signal Processing*, July 1996, (also available as Tech. Rep. USC-SIPI-278, University of Southern California, Jan. 1995.)
- [27] S. M. Kay, *Statistical Signal Processing*. Englewood Cliffs: Prentice Hall, 1993.
- [28] L. L. Scharf, *Statistical Signal Processing: Detection, Estimation and Time Series Analysis*. Menlo Park: Addison Wesley, 1991.

Part 3

Scalable Portable Parallel Algorithms for STAP Applications

Investigator: Professor Viktor K. Prasanna

Collaborator: Young Won Lim

To take advantage of High Performance Computing (HPC) platforms for radar signal processing, scalable parallel algorithms and portable implementations are essential. The algorithmic techniques must exploit the HPC architectures. These techniques will be significantly different from earlier fine grain custom VLSI approach [5, 6]. In this part, scalable portable algorithms on HPC platforms are developed for Higher-Order Post-Doppler (HOPD) STAP. This problem is computationally challenging because of the amount of computations to be performed and the need for real-time performance.

We assume a general model of High Performance Computing platforms to develop our algorithms and perform their scalability analysis. Our algorithms are written in MPI (Message Passing Interface) and they can be easily ported to various HPC platforms. In HOPD, Doppler processing (FFT computations) is followed by solving least square problems (QR Decompositions). These steps comprise most of the computation time in HOPD processing. However, the data is accessed in orthogonal directions during these steps. Instead of finding a fixed single data mapping scheme, we use two different data mapping schemes so that all the necessary data can be contained within a single processor while performing computations during these steps. In other words, by data remapping between Doppler processing and QR decomposition, we derive scalable algorithms for HOPD STAP.

The QR decomposition step causes the major computational bottleneck in HOPD STAP. If state-of-the-art parallel QR decomposition algorithms such as ScaLAPACK [7] are used, then frequent short message communications are inevitable because of sequential bottlenecks in such algorithms. Also, the communication time of the HOPD STAP implementation increases as the number of processors increases due to the collective communication patterns in the QR decomposition step. Hence, the efficiency of these algorithms becomes very low for small size matrices (less than 1000×1000) which are typically encountered in HOPD STAP. It should be noted that many QR decompositions are to be solved successively in HOPD STAP. Thus, the

cumulative parallelizing overheads of these algorithms are too excessive to be used for parallelizing HOPD STAP.

Our scalability analysis shows that the communication time of the proposed algorithm *decreases* as the number of processors increases. This is possible because as more processors are employed, smaller amount of data is allocated to each processor. As a result, the amount of data to be communicated by each processor during data remapping decreases as the number of processors increases. Also, data remapping eliminates frequent short message communications that occurs in parallel QR decomposition algorithms. Our experimental results show that additional reduction in the communication time of data remapping is possible by overlapping the communications with the computations and scheduling the communications.

We also parallelize weight computation steps of Element Space Pre-Doppler (ESPrD) and Element Space Post-Doppler (ESPdD) STAP, which are the major computational bottlenecks in the entire processing. We develop efficient distribution algorithms when a small set of processors distribute their data equally to a large set of processors. This happens frequently in the general real-time signal processing applications, because the number of sensors is limited while the number of processing nodes required for real-time needs is much greater. Also, when exploiting task parallelism, computationally intensive tasks must be distributed among many processors to alleviate computational bottleneck. Simple communication scheduling schemes will incur node contention during distribution. Thus, we develop efficient distribution algorithms by using data remapping and processor groups, which will therefore reduce the effect of node contention significantly. Although we have focused on two variations of STAP, the techniques of communication scheduling and overlapping computation with communication are adequately general. They can therefore be applied to several other signal processing computations to yield efficient implementations.

3.1 Overview

The primary goal of a radar system is to detect/track targets over a surveillance volume. The system operates by transmitting signals, receiving echoed signals from reflecting objects, and processing those signals. Radar signal processing encompasses modulation theory, detection and estimation theory, and performance evaluation. With recent improvement in computing technology, attention has been focused on building next generation radar systems. Such systems will be required to provide longer range detection of increasingly smaller targets. They also must be able to operate in the presence of hostile electronic counter measures under unfavorable environments. Thus, these systems require complicated signal processing techniques which demand more computational power than traditional techniques. STAP (Space-Time Adaptive Processing) is such technique. STAP refers to adaptive filtering algorithms that utilize both spatial and temporal information. Many variations exist in STAP algorithms. Each one has different performance characteristics and different computational complexity [1].

Several special purpose architectures have been designed for radar systems. They rely heavily on custom VLSI fine grain computations. This approach provides limited flexibility and therefore requires substantial redesign when new radar signal processing algorithms are developed. Recently, processing nodes interconnected by a high speed network become increasingly popular as HPC (High Performance Computing) platforms. Advances in networking and HPC technologies lead to the development of next generation radar Systems. Such systems are promising

because of the following characteristics. They have a scalable architecture, *i.e.* the network interface and communication network are designed to be scalable and allow processing nodes to be easily attached. Their flexibility enables a wide range of signal processing applications such as radar, sonar, acoustic, and speech processing to be performed on the same platform by simply replacing application software. They can also effectively cope with new developments in signal processing theories.

In such systems, coarse grain computation is performed by application software. The performance of such systems therefore greatly depends on the application software. It must be designed to be scalable otherwise the desired speed-up for the real-time response cannot be achieved due to overwhelming parallelizing overheads. Also, it must be designed to be portable so that it can be ported to various systems. Due to MPI standard [17], application software developed on general HPC platforms can be easily ported to new systems. In this report, issues that arise when realizing radar signal processing algorithms on general HPC platforms, will be addressed. It is necessary to develop algorithmic techniques to handle these issues, because current compilers cannot provide optimized performance. The goal of this research is providing a general framework for developing high speed, scalable, portable radar signal processing algorithms on HPC platforms. Although our research will focus on radar signal processing, it will contribute to developing algorithms for general signal processing applications as well, since these exhibit common computational characteristics. We believe our research is useful in designing next generation radar systems.

The most important requirement in radar systems is real-time response. Despite advances in computer technology, real-time response is not an easily achievable task because of the need to process larger problems (*i.e.* surveillance volume has grown rapidly). Also, the signal processing area has evolved significantly in the last decade; new theories have been developed in statistical modeling, estimation processing area has evolved significantly in the last decade; new theories have been developed in statistical modeling, estimation methods, transformation theory, among others. These sophisticated signal processing algorithms require more computation power. The computational requirements of generic radar signal processing are in the range of GFLOPS. In some STAP applications, processing 400 Giga words of data necessitates 12 Tera FLOPS. We believe that scalable algorithms and portable implementations are essential to take advantage of HPC platforms as radar systems. Algorithmic techniques such as data remapping and overlapping communication with computation must exploit the HPC architecture, which are significantly different from earlier fine grain computation.

Scalability, flexibility, programmability, and portability of HPC platforms motivate developing next generation radar systems for real-time applications such as radar systems. In the real-time applications, the guaranteed fast turn around time is the most important requirement. Both system software and application software should be designed to meet this requirement.

System software must be designed so that operating systems and message passing libraries can ensure predictable computation and communication. Unlike traditional real-time systems which depend on custom VLSI fine grain computation, coarse grain computations must be performed by application software. Application software therefore plays an important role in these systems. It should be designed to be scalable, otherwise excessive parallelization overheads prevent achieving the desired speed-up for the real-time response.

In this report, we focus on developing application software – scalable algorithms for radar signal processing on general HPC platforms. Emerging systems will retain minimal functionality

of general HPC platforms. Additionally, they will have advanced hardware and software features *e.g.* real-time OS, real-time MPI, and intelligent network interface to perform computations and communications more efficiently. Portability will also be maintained on those systems. Our developed algorithms ported onto those systems will therefore exhibit more improved performance by the advanced features of those systems.

In general, a wide range of signal processing applications such as radar, sonar, acoustic, and speech processing consist of a sequence of tasks. Each task possesses different computational characteristics. Because large amount of data is to be successively processed in real-time, following issues must be addressed.

- The computational requirement of each task is different from that of other's. Therefore, particular task can be a computational bottleneck. Scheduling the computations to allocate the computational resources becomes important.
- The data is accessed and processed in different ways in each task. It is difficult to find a data mapping which can be used through entire sequence of tasks. Even though such data mapping exists, changing data mapping between successive tasks can improve the overall performance. It is necessary to develop efficient data movement algorithms.
- Latency as well as throughput is important. Utilizing various types of parallelism in cooperation with data mapping is important.
- Latency as well as throughput is important. Utilizing various types of parallelism in cooperation with data mapping is important. Combining task and data parallelism is a possible solution for small latency and high throughput. MIMD coarse grain computation is therefore more suitable than SIMD or SIMP.

Current compiler technology cannot be used in real-time system because of following reasons. Finding optimal data mapping is known to be an *NP-complete* problem. Compilers can offer only limited way of data mapping *e.g.* block, cyclic, and block-cyclic. Loop parallelization involves large amount of unnecessary data movement between processors and increases communication time. Thus, in the this research, we will develop algorithmic techniques to handle previously mentioned issues. It will provide more optimized performance than what compilers can. We briefly list our approaches as follows.

DATA REMAPPING

In the radar signal processing, the input data cube is accessed in many ways as the processing proceeds. Thus, the design of initial data mapping as well as data remapping between tasks is needed to minimize the communication time. The data mapping problem in general terms is to design a distribution of data among the processors such that the overall communication time is minimized. Data remapping rearranges the data mapping between the tasks of the application problem to reduce the communication time.

OVERLAPPING COMMUNICATION WITH COMPUTATION

Current network interfaces and switches offer extremely low hardware latencies; however, the software overheads can be in the range of tens of micro seconds. It is necessary to exploit latency hiding techniques in the algorithm design phase. We will exploit overlapping computation with communication to hide the overheads in performing the communication, in order to minimize the communication time in data remapping.

MAPPING AND SCHEDULING OF COMPUTATION AND COMMUNICATION

Most signal processing problems have regular and known computation and communication characteristics. Exploiting the regular and known computational characteristics of signal processing algorithms on HPC platforms is needed to obtain real time performance. Because the computing power of the processors in HPC platforms are high, the mapping and scheduling should exploit coarse grain computations. To alleviate high communication overheads, communication should be carefully scheduled.

To alleviate high communication overheads, communication should be carefully scheduled.

In summary, we believe that new mapping and scheduling algorithms are needed to exploit HPC platforms for signal processing applications. These algorithms must take into account the granularity of the computations, large overheads in performing communication, and the capability to overlap computation with communication. The goal of this research is developing a general framework to develop high speed, scalable, portable signal processing algorithms on HPC platforms to support real-time needs.

In most HPC systems, lack of realistic computational models that capture the architectural features from a programmer's perspective is a major bottleneck in designing and analyzing high performance systems. Lack of analytical models severely restricts the performance of algorithms, even though they are portable. Our parallel algorithms will be developed based on our analytic model and will consider the overheads in using HPC platforms. The features of the model will be exploited in algorithm design. We will use a well known definition of scalability. A parallel algorithm is considered scalable if the execution time of the algorithm on a machine with P processors varies as $1/P$. Many HPC platforms are under development. Thus, architecture independent algorithms and portability of the developed code is very important. Specification of our algorithms using standards such as MPI will ensure portability of codes.

3.1.1 Modeling High Performance Computing Platforms

Most of the modern HPC platforms are coarse grained parallel processing systems, having a similar high level architecture. The architecture consists of three main components as shown in Figure 3.1: (a) *Powerful computing nodes*, often based on the same processors as those used in current-generation uniprocessor workstations. (b) A *low latency high bandwidth network* that interconnects the computing nodes. The network typically consists of high speed point-to-point links and routing switches. (c) *Network interfaces* that couple each processing node to the network links.

Current commercial HPC platforms offer hundreds of Giga FLOPS of computing performance. Examples of such systems are the TMC CM-5, IBM SP-2, and Cray T3D, among others. These machines have been successfully used for a variety of high performance applications with good results. The chosen model of parallel computation should accurately depict the features of the machines, including the hardware features of computation and communication units, the method used for routing, the technique employed for handling congestion, and mechanisms for synchronization. In these machines, the overheads of performing communication are still high compared with the speed of processors. These machines are therefore suitable for coarse grain computations.

For our analysis, we will model general HPC platforms as a set of high-performance SISD machines interacting through a high-speed network. Interprocessor communication is performed

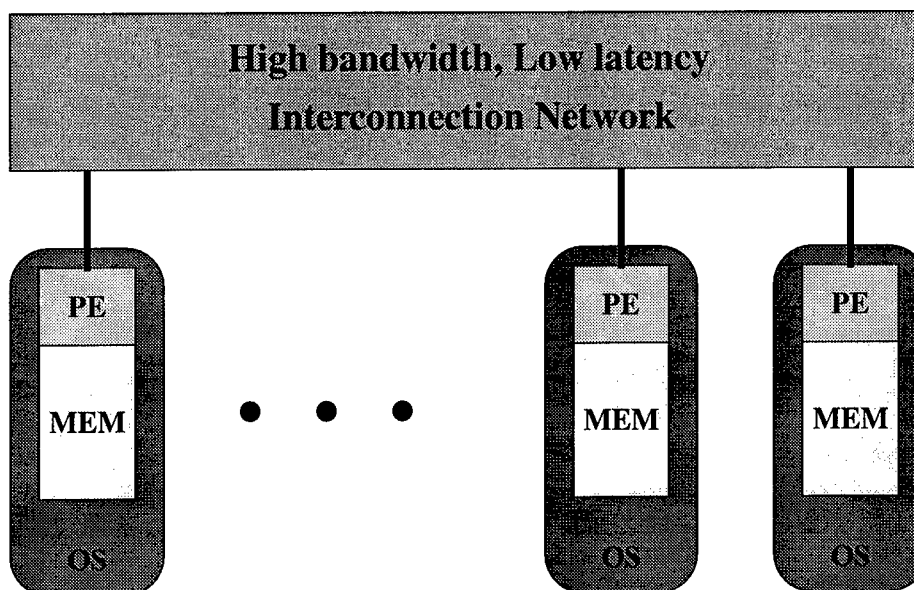


Figure 3.1: Typical HPC platform

using explicit message passing. Let T_d denote the startup time for sending a message. Let τ_d denote the transmission rate (seconds per unit of data) for data communication. The startup time, including operating system and communication protocol processing overheads, is associated with each communication step. In current generation of HPC platforms, the network (hardware) latency is very small as compared with the software overheads in message passing.

We make the following assumptions for our analysis: (1) Sending a message containing m units of data from a processor to another processor or exchanging a message of size m between a pair of processors takes $T_d + m\tau_d$ time. (2) Suppose each processor has m units of data to be routed to a single destination and the set of all destinations is a permutation, then the data can be routed in $T_d + m\tau_d$ time. (3) To perform a broadcast of a unit of data, a barrier synchronization, or a cooperative operation (sum, min, max, prefix sum etc.), $2(\log p)T_d$ time is required. Broadcasting m units of data takes $\log p(T_d + m\tau_d)$ time. (4) To perform a floating point operation, it takes t_c time.

3.1.2 Scalability

A sequential algorithm is usually evaluated in terms of its execution time, expressed as a function of the size of its input. The execution time of a parallel algorithm depends not only on input size but also on the architecture of the parallel computers and the number of processors. A parallel system is the combination of an algorithm and the parallel architecture on which it is implemented. The scalability is a measure of its ability to achieve performance proportional to the number of processors.

In the ideal case, the execution time of a scalable parallel system should linearly decrease with an increasing number of processors employed. However, in practice this is not the case, since there is a parallelizing overhead such as communication time and idle time. The purpose of scalability analysis is to predict and measure the performance of a parallel system. Several

metrics have been proposed for measuring the scalability of parallel systems.

The fixed problem analysis allows us to answer the question such as, "What is the fastest I can solve problem A on computer X?" and "What is the greatest number of processors I can utilize if I want to maintain an efficiency of 50 percent?". Large parallel computers are frequently used not only to solve the fixed size problems faster, but also to solve larger problems. This encourages a different approach to the analysis of algorithms called scaled problem analysis, whereby we consider not how efficiency varies with the number of processors, but how the amount of computation performed must scale with the number of processors to keep efficiency constant.

3.1.3 Portability and MPI

The message passing paradigm is the most generally applicable and efficient programming model for parallel machines with distributed memory. It has been used widely in parallel and distributed computing systems for some years. Message passing is attractive for portability and performance since shared memory architectures can effectively execute message passing programs whereas the reverse is not generally the case. However, the development of parallel algorithms for applications has been hindered by the absence of a standard message passing interface. The portability of library routines is becoming increasingly important as codes are required to run on several varieties of parallel computers. Portable libraries must also allow for portability across programming languages. In order to obtain maximal reuse, the same libraries must be usable from several languages. Standardization effort were begun in 1992, to define a message passing interface which would be efficiently implemented on a wide range of parallel and distributed computing systems. The main objectives of a message passing standard are portability and ease-of-use. Also the standard will provide vendors with a clearly defined set of routines that they could implement efficiently at a low level, or even provide opportunities for developing efficient hardware. It not only provides portability and ease-of-use, but also addresses to a limited extent the issues of program correctness and performance, because by providing high-level routines and abstractions, a standard can reduce the likelihood of programming errors, thereby enhancing program correctness. MPI also allows to develop parallel application libraries. The computing platforms for MPI include homogeneous and heterogeneous parallel and distributed systems and even shared memory systems.

The basic content of MPI is point-to-point communication between pairs of processes and collective communication within groups of processes. MPI also contains more advanced message passing features which allow the user to manipulate process groups, provide topological structure for process groups, and support the development and utilization of parallel libraries. Every message whether in point-to-point or collective communication has an associated data type. The primitive data types of the host language, for example `INTEGER` and `REAL` in Fortran, are supported. MPI also provides very general facilities which can be used to describe "derived" data types. The data types of MPI provide all the information required for data conversion in heterogeneous environments, and do not preclude efficient implementation of MPI in homogeneous environments. They also allow the users to send and receive messages with complicated storage patterns without the need to copy data in to and out of message buffers, and allow an implementation to optimize communications with such storage patterns.

Point-to-point and collective communications within MPI are performed within process groups. MPI defines a group as an ordered set of process identifiers, each of which is assigned a numerical rank within the group, between zero and the size of the group. Communications within MPI

are also performed within a communication context which insulates messages in different parts of the program from one another. The defining property of a context is that a message sent in one context can only be received in that same context. The communication context is the primary mechanism for isolation of messages in different libraries and the user program from one another. Process groups are user level objects in MPI but communication context are not directly visible. MPI bundles the process group and communication context concepts into a user level object called a communicator which provides communication services within a unique scope. MPI defines an initial communicator `MPI_COMM_WORLD` which has a group containing all processes of the program and a unique communication context. MPI provides routines which allow the user to dynamically create new communicators. `MPI_COMM_DUP` creates a duplicate of an existing communicator, i.e. a new communicator with the same group of processes and a different communication context. This routine is key to the construction of robust communicative parallel libraries. `MPI_COMM_SPLIT` creates one or more new communicators which contain distinct subgroups of an existing communicator and of course a different context. This routine is key to clear expression of task and control parallel programs. In the simplest use of MPI, which corresponds to a number of current communication libraries, `MPI_COMM_WORLD` is the only communicator used in the program.

The point-to-point message-passing routines form the core of the MPI standard, the basic operations being *send* and *receive*. They allow messages to be sent between pairs of processes, with message selectivity based explicitly on message tag and source process, and implicitly on communication context. Each process can execute its own code, in MIMD style, and can be sequential or multi-threaded. There is no explicit support for threads, but care has been taken to make MPI "thread safe", by avoiding the use of global state. The send receive primitives are provided in a blocking form in which the sender buffer can be reused immediately on return from send and the receiver buffer primitive `MPI_RECV`. There are four blocking send primitive corresponding to the four communication modes in MPI.

Standard The sender is blocked until the send buffer can be reused without altering the message. The receiver is blocked until the message has been copied into the receive buffer. Since the system is expected to copy the message subject to buffer resources, the send-recv pair does not guarantee synchronization. This mode seems to best represent common practice. The send primitive is `MPI_SEND`.

Synchronous The sender is blocked until the receiver issues the corresponding receive. No system buffer is required and the message can be transferred without intermediate copies, at the expense of synchronization. The send primitive is `MPI_SSEND`.

Ready-receive The program is in error if the send is issued before the matching receive has been issued. This allows a simple protocol where the message is sent "in hope" and dropped if there is no ready receive, but use demands special care. The send primitive is `MPI_RSEND`.

Buffered This mode allows the user to control the space available for buffering within a defined buffer model, providing guaranteed portability for programs that demand message buffering. The send primitive blocks until the message is copied into the buffer space or is in error if insufficient buffer space was available. The send primitive is `MPI_BSEND`.

MPI also provides primitives of the non-blocking, or immediate return, form, `MPI_ISEND` and `MPI_IRECV`, in which the message buffer must not be used until the communication has com-

pleted, similar to the immediate routines in NX/2. There is a small but comprehensive set of routines to test and wait for completion of non-blocking functions. This functionality allows the user to write programs which do not incur the overhead of copying message data into intermediate buffers.

Collective communications are provided where all processes in a process group are involved in a collective operation. A collective function is called as if it contained a group synchronization, although this property is not mandated since efficient implementations may not synchronize. The following is description of some important collective communication routines.

MPI.BARRIER Synchronization of all processes within a group.

MPI.BCAST Every process within a group receives data broadcast by a root process.

MPI.GATHER Every process within a group sends data to a root which stores the data in rank order.

MPI.SCATTER The inverse of MPI.GATHER, where a root process sends sections of data to every process within a group in rank order.

MPI.REDUCE Performs a parallel reduction over every process within a group. The operation is selected from a set of defined arithmetic and logical operators or is described as a user function. The output is available to a root process, every process, or scattered over the processes.

MPI also contains collective routines for all-to-all global communication, all-to-all personal communication also known as complete exchange, and inclusive parallel prefix also known as scan.

3.2 HOPD STAP and its Computational Requirements

The first step of HOPD STAP is Doppler processing which transforms a data cube into the Doppler domain; M -point FFT computations are performed along delay dimension for each element and each range-gate. Therefore, a total of NL M -point FFT computations are performed. The next step is weight computation. The transformed data is converted into least square problems. There are $(M-2)$ least square problems to be solved. The transformed data cube is divided into the M slabs along the range-gate dimension, which are referred to as Doppler bins as shown in Figure 3.2. The underlying matrix of each least square problem, is obtained by concatenating the three $L \times N$ size consecutive Doppler bins $i-1$, i and $i+1$ in an overlapped fashion. The resulting matrix has a size of $L \times 3N$. Each matrix is divided into two parts. The first part of the size $L_{ls} \times 3N$ defines the least square problem that is solved by QR decomposition. The solution of the least square problem *i.e.* the weight vector is applied to the remaining part (L_{wa}) in the third step. L_{ls} will typically vary from $2(3N) = 6N$ to $5(3N) = 15N$ [11].

Figure 3.2(a) shows the typical value of each dimension of a data cube (delay dimension $M=64$, element dimension $N=64$, and range-gate dimension $L=4096$ ($L_{ls}=1024$) for a nominal size problem. All these parameters can change for a bigger size problem). In the Doppler processing stage, NL M -point FFT computations are performed. In the weight computation stage, there are $(M-2)$ QR decomposition computations of size $L_{ls} \times 3N$. It is noted that only $L_{ls}=16N(=1024)$ parts of range-gates are used for the weight computation, as shown in Figure 3.2(b). The weight application and detection stages have also many independent problems, but

the amount of computations is comparable to that of Doppler processing stage. The sequential m -point FFT problem requires $m \log m$ complex multiplications and additions. Assuming one complex multiplication and addition operation take 5 *flops**, the sequential m -point FFT takes $5m \log m$ *flops*. The sequential QR decomposition of a $m \times n$ matrix needs $n^2(m - n/3)$ *flops* [14]. Therefore, the estimated computational requirement of HOPD STAP is as follows.

- Doppler Processing: NL M -point FFT Problems
 $(NL) \times 5 \times M \log M = 480 \times 2^{20} \text{ flops} = 480M \text{ flops}$
- Weight Computation: $(M-2)$ QR Decomposition Problems of size $16N \times 3N$
 $(M - 2) \times \{(3N)^2 \times (16N - 3N/3)\} = 2100M \text{ flops} \approx 2G \text{ flops}$

The entire HOPD STAP requires approximately $3G$ *flops*. The real time requirements vary by the application. Suppose that the entire set of steps is to be computed in less than 1 second, then at least 54 POWER2 processors (operating at 66MHz) are needed for the real time computation of HOPD STAP in an ideal case.

3.3 Scalable Parallel Algorithms for HOPD STAP

In this section, we first discuss the characteristics of general HPC platforms for algorithm development and scalability analysis. Then, we develop techniques for data remapping, scheduling and overlapping communication with computation to obtain scalable parallel algorithms for HOPD STAP.

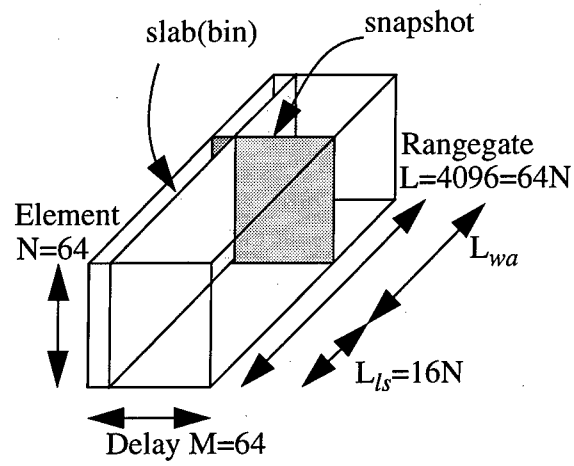
3.3.1 Computational Model

For our analysis, we will model the general HPC platforms as a set of high-performance SISD machines interacting through a high-speed network. Assuming MPI is used, interprocessor communication is performed using explicit message passing [8]. Let T_d denote the startup time for sending a message. Let τ_d denote the transmission rate (seconds per unit of data) for data communication. The startup time, including operating system and communication protocol processing overheads, is associated with each communication step. In current generation of HPC platforms, the network (hardware) latency is very small as compared with the software overheads in message passing.

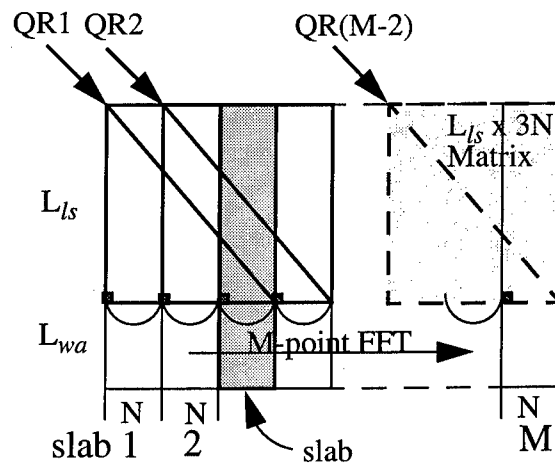
We make the following assumptions for our analysis: (1) Sending a message containing m units of data from a processor to another processor or exchanging a message of size m between a pair of processors takes $T_d + m\tau_d$ time. (2) Suppose each processor has m units of data to be routed to a single destination and the set of all destinations is a permutation, then the data can be routed in $T_d + m\tau_d$ time. (3) To perform a broadcast of a unit of data, a barrier synchronization, or a cooperative operation (sum, min, max, prefix sum *etc.*), $2(\log p)T_d$ time is required. Broadcasting m units of data takes $\log p(T_d + m\tau_d)$ time. (4) To perform a floating point operation, it takes t_c time.

For currently available HPC platforms, the ratio of T_d to t_d is in the range of several hundreds to few thousands as shown Table 3.1[†]. To reduce the communication time, the high startup cost as well as the message length, must be considered.

*C.B. Moler's concept of a *flop* is used. One *flop* roughly constitutes the effort of doing a floating add, a



(a)



(b)

Figure 3.2: Data cube and FFT & QRD computations.

Table 3.1: Communication features of various HPC platforms.

Machine	$T_d(\mu sec)$	$\tau_d(\mu sec/byte)$	T_d/τ_d
T3D	93	0.043	2163
SP-2	46	0.035	1314
CM-5	86	0.12	716
Paragon	82	0.26	315
iPSC/860	60	0.50	120
iPSC/2	700	0.36	1944

3.3.2 Scalability and Portability

We use a standard definition of scalability and efficiency [13, 15]. A parallel algorithm is considered scalable if the execution time of the algorithm on a machine with p processors varies as $\frac{1}{p}$. Efficiency is defined as $\frac{T_s}{p \cdot T_p}$, where T_s is the sequential execution time and T_p is the parallel execution time. Our goal is to design scalable algorithms which provide high-speed execution on available partition sizes of machines for problem sizes useful to radar signal processing community. In our analysis, we consider machine sizes that are not “large”, for example, at this time, the largest installed accessible coarse grain machines have less than 1K nodes.

Our implementations were performed using the MPI standard, and therefore can be easily ported to other High Performance Computing (HPC) platforms. Our choice of MPI allows us to access many communication features provided by emerging HPC platforms. At this time, we have implemented our algorithms on SP-2 and T3D.

3.3.3 Data Remapping, Scheduling, and Overlapping Computation with Communication

For a scalable performance, efficient data mapping and data distribution schemes should be developed so that the parallelizing overhead is minimal; these schemes should minimize the inter-processor communication time. Our algorithm exploits these schemes and is scalable in the range of $p \leq M$.

A. Data Remapping

We choose data mapping schemes which allow a processor to contain all the necessary data for computing each task. In our approach, tasks such as FFT and QR decomposition do not need interprocessor communication. In the Doppler processing stage, the data cube is accessed along the delay dimension. However, in the weight computation stage which follows Doppler processing stage, the data cube is accessed along the range-gate dimension which is orthogonal to the delay dimension. Thus, different data mapping schemes are necessary for these computation steps. The remapping process refers to the transformation of the initial data mapping into another data

floating point multiply, and a little subscripting [14].

[†]Some results were obtained by our own experiments. Others are from Technical Report No. UCB/CSD92 #675, University of California, Berkeley. It is assumed that MPI is used for T3D.

mapping. Our scalability analysis in the following section shows that our approach using data remapping exhibits scalable performance characteristics, which allows each processor to compute sequentially as much as possible without interprocessor communication by providing the data required in its computation.

Initially, the data cube is partitioned along the range-gate dimension into blocks containing L_{ls}/p snapshots and the processors are assigned to these partitioned data in a cyclic distribution as shown in Figure 3.3(a). We call this data mapping as *snapshot mapping*. Each processor has $\frac{4L_{ls}MN}{p}$ size data. $\frac{4L_{ls}N}{p}$ M -point FFTs can be performed in each processor without interprocessor communication. The other mapping is shown in Figure 3.3(b), which we name as *slab mapping*. Here, the L_{ls} part of slabs (Doppler bins) is partitioned along the delay dimension among the p processors. Note that only L_{ls} part of the data cube is used for weight computation and the rest (L_{wa} part) is used for weight application. The L_{wa} part of the data cube is not remapped. The size of the data which will be remapped in each processor, is $\frac{L_{ls}MN}{p}$. Each processor sends $\frac{L_{ls}MN}{p^2}$ size data to all the processors and receives $\frac{L_{ls}MN}{p^2}$ size data from all the other processors as shown Figure 3.3(c).

Each processor can form $(M/p - 2)$ matrices of size $L_{ls} \times 3N$ for computing the weight vectors. Therefore, a total of $(M/p - 2)$ QR decomposition problems can be solved without communication and two more problems can be solved by communicating with adjacent processors. Exchange of boundary data between the adjacent processors is needed. This is shown in Figure 3.4. This data exchange is performed during the remapping process.

B. Scheduling

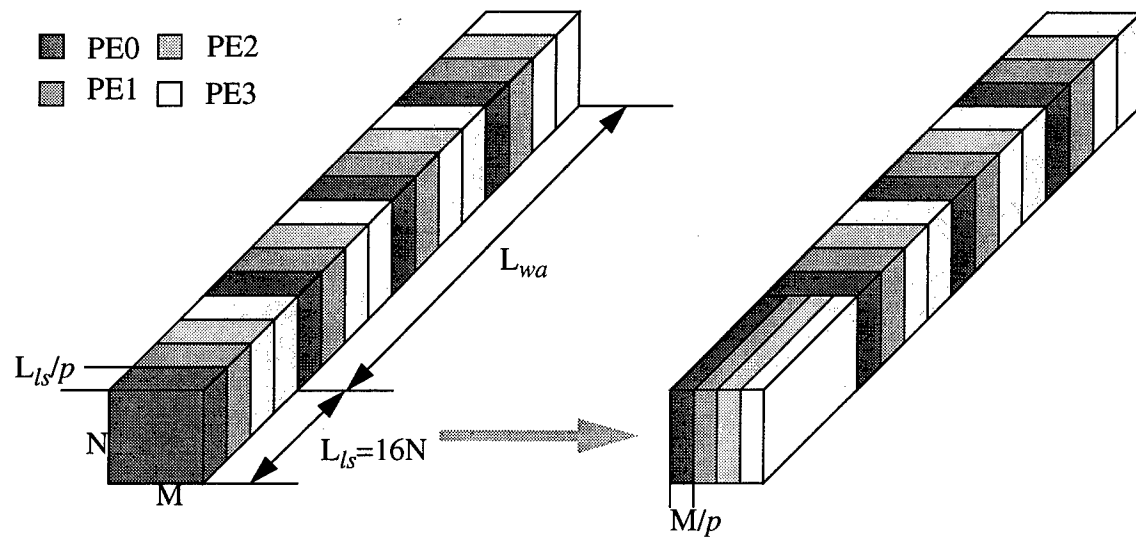
As mentioned before, HOPD STAP consists of a sequence of tasks. The input to a task is output of the previous task. Due to the data dependency and different data access patterns between tasks, interprocessor communication is required. For scalable performance, a good scheduling scheme is necessary. The QR decomposition tasks are dependent on only L_{ls} part of the data cube which were Doppler processed. The output of QR decomposition tasks are used along with the rest of the Doppler processed data for detection.

Also, the scheduling of communication is important to fully exploit the available communication bandwidth and to avoid node contention. During the remapping process, every processor distributes equal size data to all the other processors. It is a collective communication but it can be converted into $(p-1)$ steps of all-to-all personalized communications (*i.e.* permutations) by a simple scheduling. In each of the $(p-1)$ communication steps, each processor sends equal size data to another processor and receives equal size data from another processor. This exploits the fact that total p point-to-point communication can be performed in parallel among p processors.

By employing a butterfly-like communication pattern, the number of communication steps can be reduced to $\log p$ and the startup overhead can be reduced by a factor of $\frac{p-1}{\log p}$. By using features such as *communication group* and *context* in MPI, the communication between processors can be localized. However, in this case, there is an additional overhead in buffer copying.

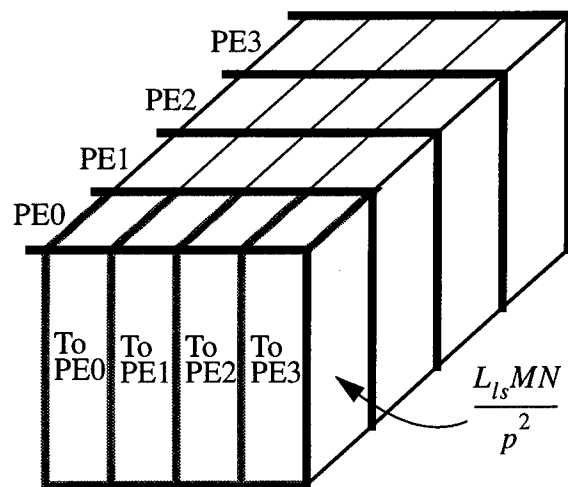
C. Overlapping Communication with Computation

We reduce the communication time of data remapping by scheduling the communications and by overlapping the communication with the computation. Using asynchronous non-blocking communication primitives provided by the MPI standard (such as `MPI_Isend()`, `MPI_Irecv()`, and



(a) Snapshot Mapping

(b) Slab Mapping



(c) Data to be communicated

Figure 3.3: Data remapping process.

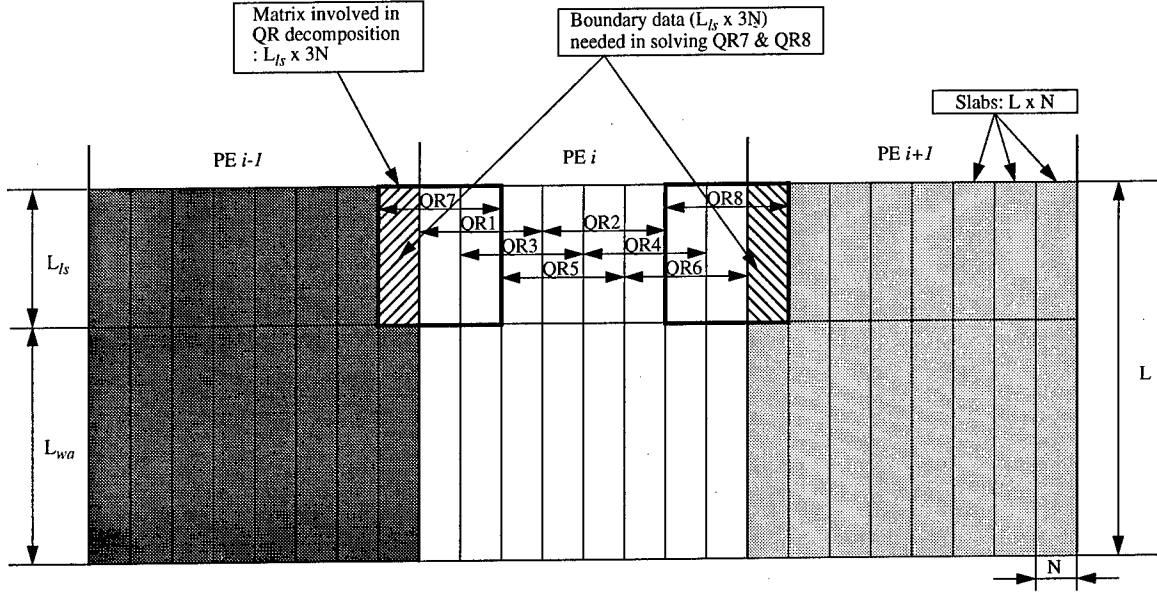


Figure 3.4: Boundary data exchange.

`MPI_Waitall()`), we can overlap the communication needed for the remapping process with the FFT computations. First, each processor performs sequential FFT computations over L_{ls} portion of the data cube. As soon as these $\frac{L_{ls}N}{p}$ M -point FFT computations are computed, the remapping process can start. Asynchronous non-blocking commands `MPI_Isend()` and `MPI_Irecv()` return immediately after initiating communication without waiting for completion. Therefore, the rest of the FFT computations over L_{wa} part of the data cube can be performed by processors, while the actual data transfer is handled by communication coprocessors.

A processor can solve one QR decomposition problem, after receiving three consecutive slabs from the remapping process. Suppose the data cube is partitioned along the delay dimension by K such that $\frac{M}{pK} \geq 3$. The remapping process is divided into K ($p-1$) permutation steps. Then, the computation of QR decomposition can be also overlapped with the communication.

D. Scalability analysis

For scalability analysis, we assume that communication is not overlapped with computation. The experimental results of overlapping the remapping process with the computation are shown in Section 3.6. The remapping process consists of ($p-1$) permutations. During each permutation, every processor sends and receives $\frac{L_{ls}MN}{p^2}$ size data. Therefore, the communication time for the remapping process is

$$T_{remap} = (p-1) \left\{ T_d + k \cdot \frac{L_{ls}MN}{p^2} \cdot \tau_d \right\} \quad (3.1)$$

where k is the size of a data point in bytes. Boundary data exchanges involve two additional communications with message size of $k \cdot L_{ls}N \cdot \tau_d$. The communication time for boundary data exchange becomes $2\{T_d + k \cdot L_{ls}N \cdot \tau_d\}$. Also, for the weight application, the solutions of QR

decompositions, i.e. the weight vectors must be permuted among all the processors. This involves $(p-1)$ all-to-all communications with message size of $k \cdot \frac{M}{p}$. It takes $(p-1)\{T_d + k \cdot \frac{M}{p} \cdot \tau_d\}$ communication time. Because $L_{ls}N \leq \frac{L_{ls}MN}{p}$ and $\frac{M}{p} \ll \frac{L_{ls}MN}{p}$ when $p \leq M$, the total communication time of our approach is *upper bounded* by

$$T_{comm} \leq 2p \cdot T_d + k \cdot \frac{3L_{ls}MN}{p} \cdot \tau_d \quad (3.2)$$

It is noted that $2p \cdot T_d \ll k \cdot \frac{3L_{ls}MN}{p} \cdot \tau_d$, for the values taken by p ($\leq M$), L_{ls} ($= L/4$), M , and N in typical STAP applications. For example, $M=64$, $N=64$, $L_{ls}=1024$, and $p=32$, the startup overhead $2p \cdot T_d=2.9$ msec which is much smaller than $k \cdot \frac{3L_{ls}MN}{p} \cdot \tau_d=110$ msec for the SP-2. Therefore, T_{comm} can be approximated as

$$T_{comm} \approx k \cdot \frac{3LMN}{4p} \cdot \tau_d \quad (3.3)$$

In summary, the total startup overheads are $2p \cdot T_d$ and total amount of data to be communicated is approximately $k \cdot \frac{3LMN}{4p} \cdot \tau_d$ when data remapping is used. If data remapping is not used, then $6(M-2)N$ communication steps are required to solve $(M-2) \times 3N$ size QR decompositions. (Solving one $m \times n$ size QR decomposition requires $2n$ communication steps.) In this case, the startup overhead will be $6(M-2)N \cdot T_d$. Thus, remapping which eliminates frequent short message communications, reduces startup overheads. Also, it reduces total amount of message to be communicated. This will be analyzed in detail in the Section 3.5.

The sequential execution time T_s consists of time for executing LN M -point FFTs, $(M-2) \times 3N$ size QR decompositions, weight application, and detection. Weight application and the rest of the computations including detection are not computationally demanding. Thus, T_s becomes

$$T_s = \{9(M-2)N^2(L/4 + N) + 10LMN \log M\} \cdot t_c \quad (3.4)$$

Due to the remapping of data, the load on the processors is well balanced. In other words, the FFT and QR decomposition computations are equally distributed among the processors. Therefore, the parallel execution time is

$$\begin{aligned} T_p &= \frac{T_s}{p} + T_{comm} \\ &= \{9(M-2)N^2(L/4 + N) + 10LMN \log M\} \cdot \frac{t_c}{p} + k \cdot \frac{3LMN}{4p} \cdot \tau_d \end{aligned} \quad (3.5)$$

It should be noted that $T_{comm} \ll \frac{T_s}{p}$. In other words, the computation time ($\frac{T_s}{p}$) of each processor is much larger than the communication time (T_{comm}). Thus, the proposed algorithm is scalable in the sense that the execution time of the algorithm on a machine with p processors varies as $\frac{1}{p}$ in the range of $p \leq M$. Efficiency becomes

$$E = \frac{T_s}{p \cdot T_p} = \frac{T_s}{p(\frac{T_s}{p} + T_{comm})} = \frac{1}{1 + \frac{T_{comm}}{T_s/p}} \quad (3.6)$$

Because $T_{comm} \ll \frac{T_s}{p}$, near ideal efficiency is achieved.

3.4 Comparison with previous approaches

A previous approach for HOPD processing uses row wrap mapping scheme in order to avoid data redistribution (remapping) [11]. In their approach, FFT computations can be performed sequentially without inter-processor communication. However, parallel QR decomposition is needed to exploit their row wrap mapping scheme. The parallel QR decomposition algorithm developed by Whitman *et al.* [12] was used in [11]. In [11], QR decomposition comprises most of the computation as well as communication time. Their experimental results show that its communication time increases rapidly as the number of processors is increased. When a 32 node SP-1 is employed, the percentage of communication time is more than 50% of the total execution time.

The parallel QR decomposition algorithm of [12] and its timing diagram for the first two steps are shown in Figure 3.5 and Figure 3.6. Processors are idle between PASS and SCALE steps and the processors communicate during PASS and BRDCAST steps. The communication pattern is collective for PASS and is broadcast for BRDCAST. PE0 (which contains the main diagonal element a_{ii}), can start SCALE step only after it has received the accumulated dot products from the other PE's. During the SCALE step, all the processors except PE0 remain idle. Also, after receiving the broadcast scale factors from PE0, all the processors can perform UPDATE. The estimated computation and the communication times are

$$t_{comp} = \{(n^2 + 2n) \cdot \frac{m}{p} + n^2 + 5n\} \cdot t_c \quad (3.7)$$

$$t_{comm} = (p + \log p - 1) \{n \cdot T_d + k \cdot (\frac{n^2}{2} + \frac{3n}{2}) \cdot \tau_d\} \quad (3.8)$$

where m and n ($m \geq n$) represent the row and column size of the matrix. t_c is the time for performing one floating point operation, and T_d and τ_d are startup time and per-word transfer time, respectively. From Equation (3.8), the communication time increases in proportion to p , the number of processors. There is another parallelizing overhead: there exists sequential bottleneck where only one processor is active and all the others are idling. The idle time of SCALE step is estimated in Equation (3.9) and during this time, $(p-1)t_{idle}$ useful work could be done otherwise.

$$t_{idle} = \frac{3}{4}(n^2 + 7n) \cdot t_c \quad (3.9)$$

To solve one QR decomposition problem, there are n communication steps and each step i involves collect and broadcast communications using message of size $(n-i+1)$, where n represents the column size of the matrix involved. Therefore, a total of $\sum_{i=1}^n 2(n-i+1) = n(n+1)$ messages are communicated. Because there are $(M-2)$ QR decomposition problems of size $L_{ls} \times 3N$ in the HOPD STAP, the total size of the messages to be communicated is $3N(3N+1)(M-2)$. This is almost $\frac{1}{7}$ -th of the entire data cube size we are considering. By assuming that all these data can be sent in two steps (*i.e.* collect and broadcast communications) by ignoring communication startup overheads and idle times, a *lower bound* on the communication time of the algorithm in [3] can be obtained. Using the time for point-to-point communication time as $(T_d + m \cdot \tau_d)$ and the time for broadcasting as $(T_d + m \cdot \tau_d) \log p$, the communication time T'_{comm} in [11] is *lower bounded* as

$$T'_{comm} \geq (p + \log p - 1) \{T_d + k \cdot \frac{3}{2} N(3N+1)(M-2) \cdot \tau_d\} \quad (3.10)$$

$$\approx (p + \log p) (k \frac{9}{2} MN^2) \cdot \tau_d \quad (3.11)$$

The i -th step of the parallel QR decomposition based on Householder transformation ($i = n, \dots, 1$ where n is the column size of the underlying matrix):

$$u_i = a_i + \gamma_i e_i, \quad \gamma_i = \text{sign}(a_i) \cdot \sqrt{a_{i,i}^2 + \dots + a_{i,n}^2},$$

$$\pi_i = \frac{1}{(a_i + \gamma_i) \cdot \gamma_i}, \quad A \leftarrow \left(I - \pi_i u_i u_i^T \right) \cdot A$$

- (1) **DOTPRD**[i]: Compute the dot products of the local block of column i with columns i, \dots, n . Only use elements that are in matrix rows i through m .
- (2) **ACCUM**[i]: Wait for the $n-i+1$ dot products from a successor processor, if there is one, and add on the local dot products.
- (3) **PASS**[i]: Send the revised $n-i+1$ dot products to a preceding processor, if there is one.
- (4) **SCALE**[i]: The processor which contains a_{ii} is the last to receive the revised dot products. It performs a square root operation to determine the scale factor g_i , adds it to the main diagonal element, corrects the inner products for the addition of g_i to the i -th element of u_i , and multiplies them by p_i .
- (5) **BRDCAST**[i]: Wait for the $n-i+1$ scale factors from a preceding processor, if there is one, and pass them on to a successor processor, if there is one.
- (6) **UPDATE**[i]: Update the local block of column i through n by subtracting multiples of the local block of column i .

Figure 3.5: The parallel QR decomposition algorithm.

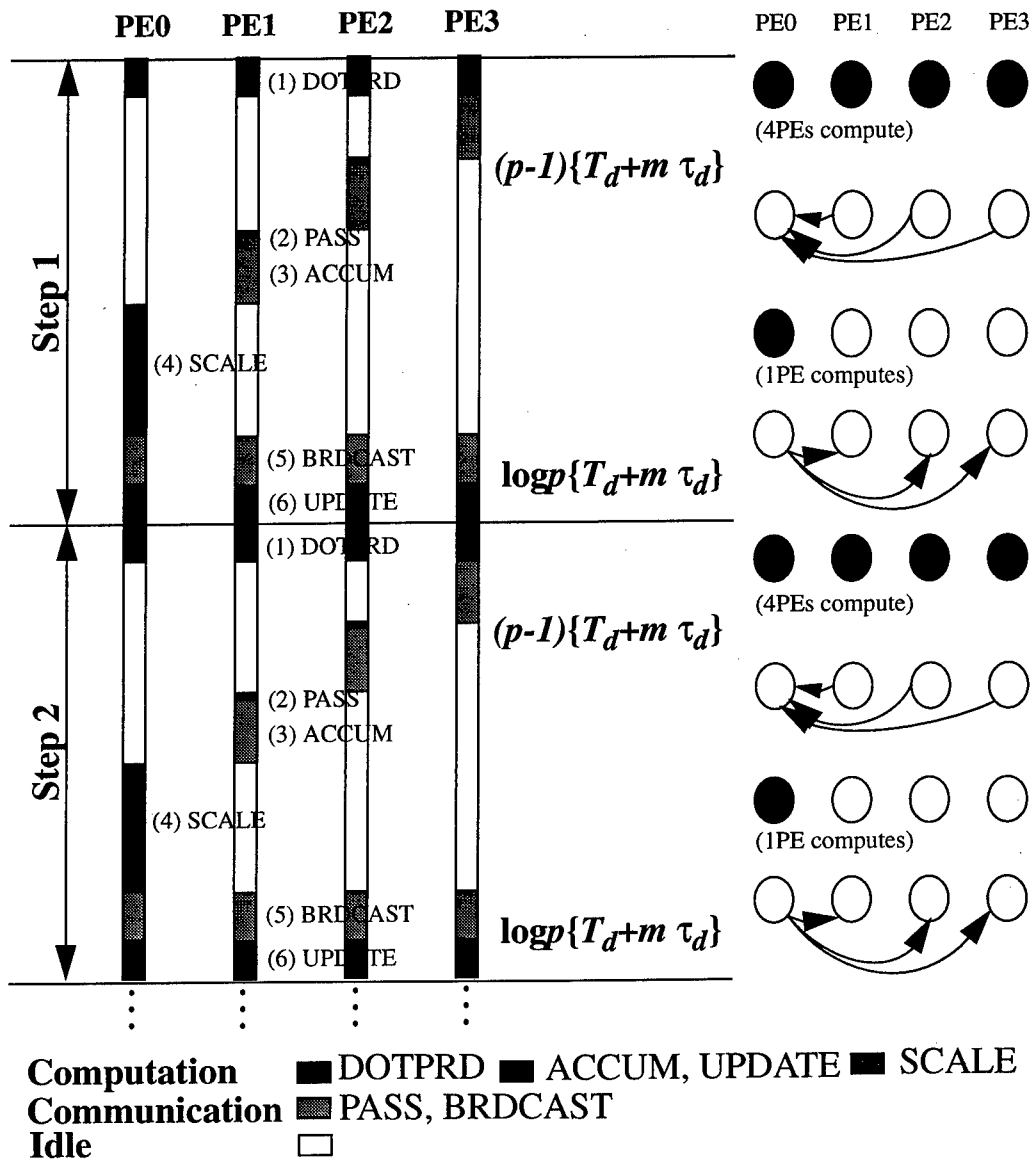


Figure 3.6: The timing diagram of the parallel QR decomposition algorithm for the first two steps. Four processors are assumed for the sake of illustration.

In other words, actual communication time of [11] will take at least $(p + \log p)(k \frac{9}{2} MN^2) \cdot \tau_d$. Consider the case of $M=64$, $N=64$, $L_{ls}=1024$, $k=8$, and $p=32$, then the communication time in the our algorithm is $T_{comm} \approx 384Kword \cdot \tau_d$, and the communication time in [11] becomes $T'_{comm} \geq 40000Kword \cdot \tau_d$. This explains why the efficiency of the algorithm in [11] drops drastically as the number of processors is increased. It should be also noted that there is significant processor idle time as shown in Equation (3.9). In contrast, the communication time of the proposed algorithm is inversely proportional to p as shown in Equation (3.3). If more processors are used in the range $p \leq M$, the communication time becomes less. Figure 3.7 shows the estimated communication time of the proposed algorithm and the communication time in [11] for the SP-2. As shown in Figure 3.7, $T_{comm} \ll T'_{comm}$. It is also observed that T_{comm} decreases while T'_{comm} increases with increasing number of processors.

ScaLAPACK [7] also provides QR decomposition routine. It uses BLAS(Basic Linear Algebra Subroutine), PBLAS (Parallel extension of BLAS), and BLACS(Basic Linear Algebra Communication Subprogram). Because ScaLAPACK employs block cyclic data mapping, data redistribution is necessary between FFT computations and QR decompositions. The efficiency of ScaLAPACK drops significantly unless large size matrices are used. For example, it can deliver about sustained 21 GFLOPS on 16×32 node Paragon for the matrix size of 35000×35000 . However, if the matrix size becomes less than 1000×1000 , then less than 1 GFLOPS can be achieved on Paragon. This means that the parallelization overheads, *i.e.*, communication and idle times dominate the computation time. Because the typical matrix size for HOPD STAP is $16N \times 5N$ (1024×320 when $N=64$), we cannot expect high efficiency by using ScaLAPACK.

3.5 Experimental Results

We have performed our implementations using C and MPI[†] on the SP-2 at Maui High Performance Computing Center and on the T3D at the Pittsburgh Supercomputing Center. Optimized versions of code provided by ESSL (Engineering Scientific Subroutine Library) were used in our SP-2 implementations.

Remapping process can be performed either by using asynchronous non-blocking communication or by using synchronous blocking communication as shown in Figure 3.8. Asynchronous non-blocking communication allows the communication in the remapping process to be overlapped with L_{wa} parts of FFT computations. In this case, the actual time spent in communication activity becomes $\sum_{i=1}^{p-1} t_{is}(i) + t_{ir}(i) + t_{wt}(i)$, where $t_{is}(i)$, $t_{ir}(i)$, and $t_{wt}(i)$ are time spent in the i -th `MPI_Isend`, `MPI_Irecv`, and `MPI_Waitall`, respectively. In the case of non-blocking communication, communication time becomes $\sum_{i=0}^{p-1} t_{sr}(i)$, where $t_{sr}(i)$ represents time that takes in the i -th `MPI_Sendrecv`. Table 3.2 shows the time results of FFT and data remapping process. It should be noted that message size to be communicated decreases as the number of processors increases. Thus, the communication time of remapping process decreases as the number of processors increases up to M . Figure 3.9 illustrates the reduction in the remapping time by overlapping the remapping process with the computation. On both machines, the overlapped remapping process takes less than 37% of the non-overlapped one.

SP-2 was configured in the dedicated mode to obtain the results of the HOPD STAP implementation shown in Table 3.3. For the sake of comparison, an earlier implementation [11] of the same technique using the same parameters runs in about 30 seconds on a 32 node SP-1.

[†]MPICH version was used.

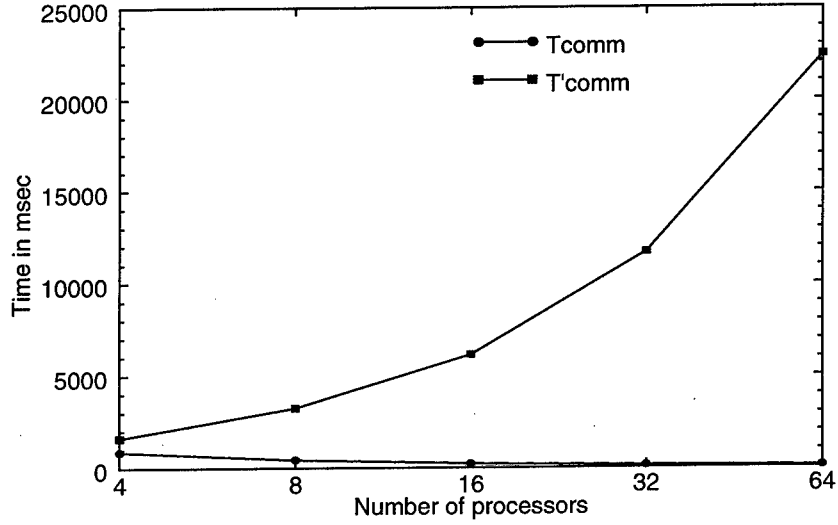
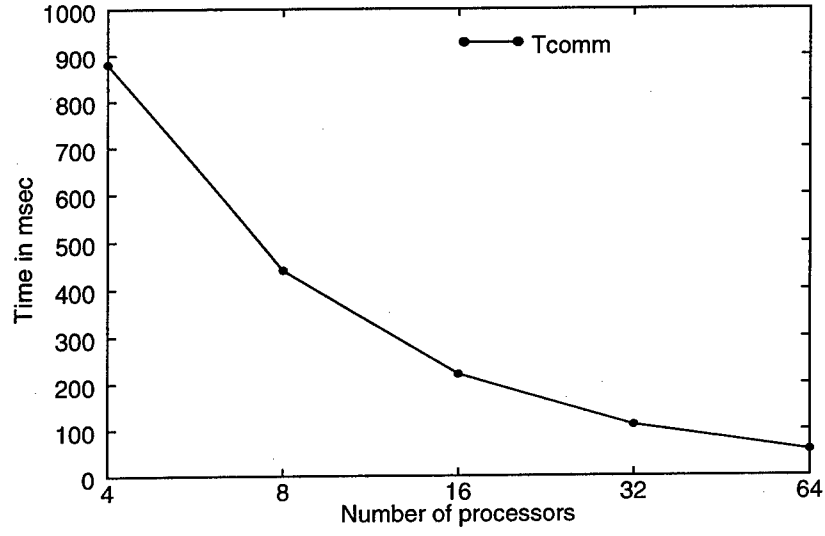
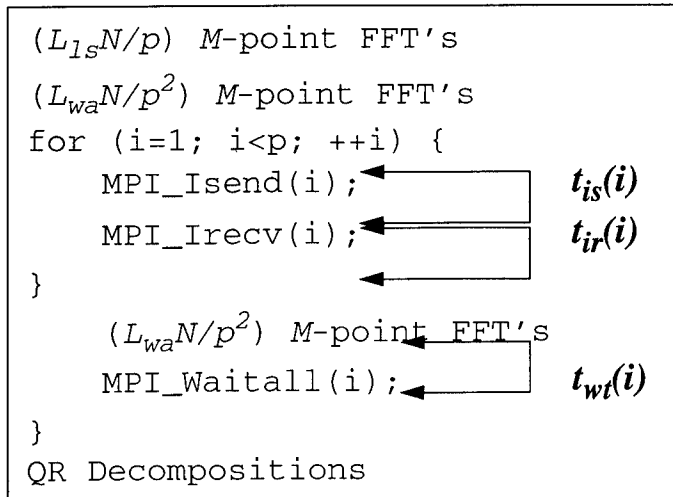
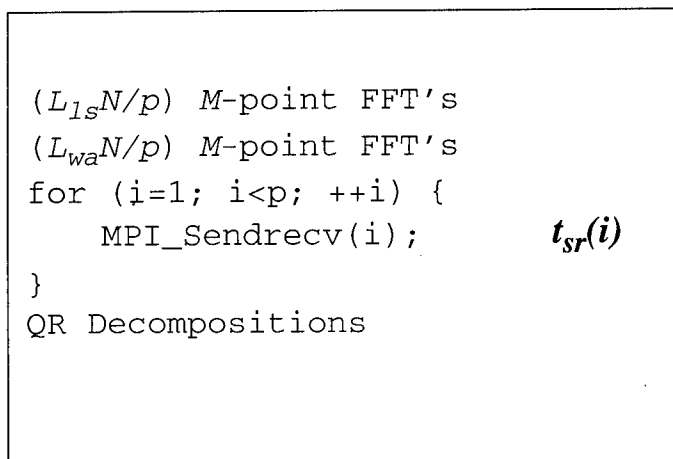


Figure 3.7: Estimated T_{comm} and T'_{comm} on SP-2 for $M=64$, $N=64$, $L_{ls}=1024$. Each data is assumed as single precision complex number (8 bytes).



(a) Asynchronous Nonblocking communication



(b) Synchronous Blocking communication

Figure 3.8: Asynchronous Nonblocking communication v.s. Synchronous Blocking communication for Remapping.

Table 3.2: Timing Results for computing FFT and Data Remapping on T3D and SP2.

Timing Results for Performing FFT& REMAPPING on T3D (msec)^a

	Number of PEs	$p=4$	$p=8$	$p=16$	$p=32$	$p=64$
Doppler Proc	Number of 64-point FFT's per PE	65,536	32,768	16,384	8,192	4,096
	Total time for FFTs	12800.0	6420.0	3210.0	1610.0	200.0
	Message size	2MB	512KB	128KB	32KB	8KB
Data Remap	($p-1$) Async Nonblocking pt-to-pt communications	63.5	37.6	21.8	15.1	11.8
	($p-1$) Synchronous blocking sendrecv communications	264.0	154.0	84.5	48.1	32.0

a. Cray Library was not available. Numerical Recipes routine was used.

Timing Results for Performing FFT& REMAPPING on SP2 (msec)^b

	Number of PEs	$p=4$	$p=8$	$p=16$	$p=32$	$p=64$
Doppler Proc	Number of 64-point FFT's per PE	65,536	32,768	16,384	8,192	4,096
	Total time for FFTs	1730.0	858.0	428.0	214.0	106.9
	Message size	2MB	512KB	128KB	32KB	8KB
Data Remap	($p-1$) Async Nonblocking pt-to-pt communications	64.7	37.5	13.2	8.57	6.71
	($p-1$) Synchronous blocking sendrecv communications	351.0	220.0	101.09	43.99	29.8

b. IBM ESSL library was used for computing FFT. MPICH was used. Loadleveller data.

Table 3.3: Total Execution Times on SP-2 for $M=N=64$, $L=4096$.

No. of PE's	$p = 1$	$p = 4$	$p = 8$	$p = 16$	$p = 32$	$p = 64$
SP-2	55sec	15 sec	7.8 sec	3.9 sec	2.0 sec	0.96 sec

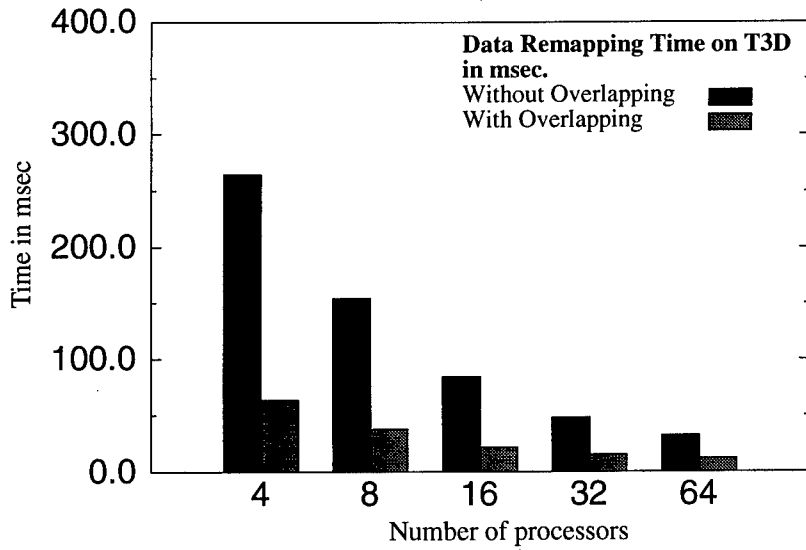
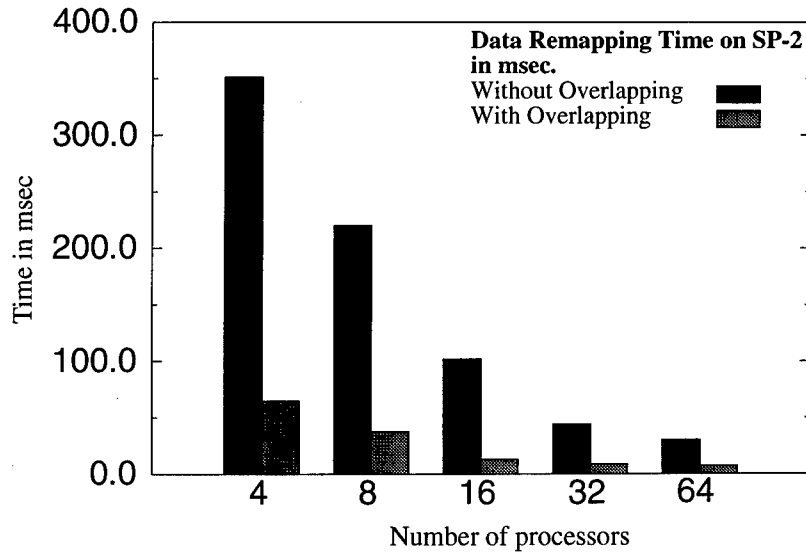


Figure 3.9: Experimental results of data remapping with overlapping for $M=64$, $N=64$, $L_{ls}=1024$ on SP-2 and T3D. Each data is assumed as a single precision complex number (8 bytes).

3.6 A Methodology for the Design of Scalable Solutions

Radar signal processing is typical of the applications that arise in embedded real-time environments. In this section, we highlight a coarse-grained parallel approach to Space Time Adaptive Processing (STAP), a class of radar signal processing. We first describe a model of the target HPC platforms. We then perform scalability analysis on STAP algorithms, and demonstrate the design of efficient parallel algorithms for these tasks.

Our approach to design efficient solutions for real-time applications on the HPC platforms consists of the following steps:

- Definition of a realistic computational model of the HPC systems. The model represents the salient features of the underlying hardware architecture.
- Analysis of the application task to be performed, and identifying the phases that are the most time consuming.
- Quantification of the overheads incurred in executing the task on the architecture, with the assistance of the model. The overheads and the computation time are calculated as a function of the number of processors.

These functions are then used to study the scalability of the application-architecture pair. If the total execution time (overheads + computation time) is inversely proportional to the number of processors, then the solution is said to be scalable.

For our analysis of STAP tasks, we define a General purpose Distributed Memory (GDM) model. This model balances simplicity, accuracy and generality. We model the parallel system as a set of processors, interacting through a high speed network. Most HPC systems have a uniform high-level architecture of this form.

The communication operation in such a system can be modeled with two parameters. One is *startup time* which occurs in every message transfer. This includes software and communication protocol processing overheads. The other is *unit transmission time* which is the cost of transferring a message of unit length. Let T_d denote the startup time and τ_d denote the transmission time. Then, a communication operation for sending a message containing m units of data from a module to another module takes $T_d + m\tau_d$ time. This is the case when “blocking” send is used and no special hardware for communication is available. During $T_d + m\tau_d$ time, the sending processor is busy handling the communication. However, if there is a communication processor and nonblocking send is used, the sending processor has to spend only the startup time to activate the communication processor. After this initial startup time, the processor is free and can continue its computations, while actual transfer of messages is handled by the communication processor.

The GDM model accurately represents the communication features of many HPC architectures. The model also represents features for overlapping computation with communication and latency hiding (for example, this feature represents the DMA access capability of the LANai network interface chip in Myricom systems [16]). In this case, a communication operation is assumed to take only T_d time.

Since many different platforms are under development, architecture independent algorithms and portability of the developed code are very important. We specify our algorithms using the MPI standard [17] to ensure portability of our code.

In general, the data layout problem is to design a distribution of data among the processors such that the overall communication time is minimized, and the workload is balanced. In typical

signal processing applications, the data is accessed and processed in different ways in each sub-problem. The goal of data remapping is to rearrange the data between the steps of the algorithm so as to reduce the communication time. In some scenarios, the overhead incurred in performing the remapping operation might be larger than the benefit due to remapping. To estimate the tradeoffs, we use the model to evaluate the benefits and overheads due to data remapping.

Most signal processing problems have regular and known computation and communication characteristics. This permits mapping and scheduling of computations to be performed at compile time. Indeed, this feature has been exploited in systolic arrays to transform the computations into a form that can be executed with localized communication. The scheduling problem for HPC platforms is different from that of custom VLSI systems on account of the larger granularity of the computations, large overheads in performing communication, and the capability to overlap computation with communication.

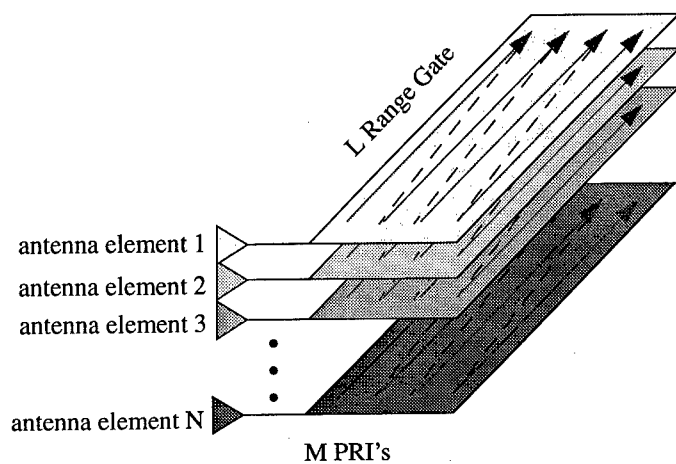
3.7 Computational Characteristics of Element-Space STAP

STAP consists of an antenna array (with N elements) that transmits pulses and then receives the reflected pulses. Each element accumulates data in this manner at L different ranges. Thus the total data collected in one PRI (Pulse Repetition Interval) has NL data samples. After M PRI's, the accumulated data constitute a Coherent Processing Interval (CPI) data cube. Figure 3.10 (a) shows the data arrival sequence, while Figure 3.10 (b) shows the data cube.

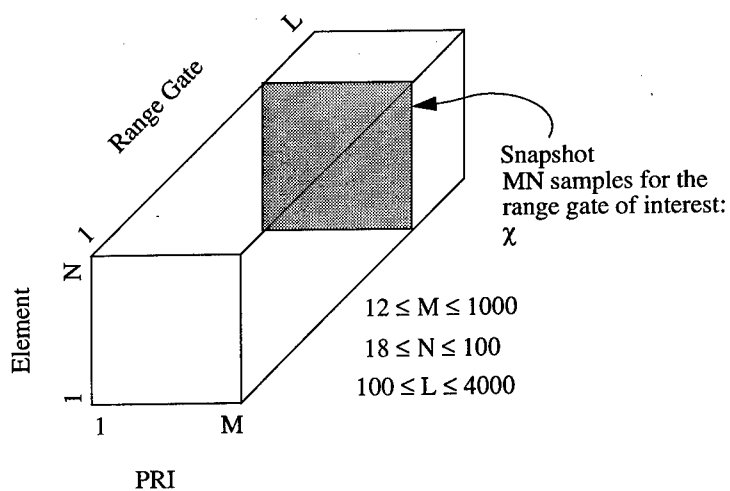
A significant portion of STAP involves matrix algebra computations. STAP consists of three major stages – training strategy, weight computation, and weight application. Generally, weight computation requires the solution of a linear system of equations. This step is a computation intensive portion of the space time processor. Due to enormous computational requirements of fully adaptive STAP, many partially adaptive processing techniques are developed. In this report, we are considering on ESPrD (Element-Space Pre-Doppler) and ESPsD (Element-Space Post-Doppler) processing techniques, which are both partially adaptive STAP approaches.

We parallelize weight computation steps of ESPrD and ESPsD, which are the major computational bottlenecks in the entire processing. In the ESPrD, $M' = M - K + 1$ ($K = 2 \sim 3$) adaptive processing are performed over subsets of size $N \times K$ of a snapshot, as shown in Figure 3.11. The data cube is accessed along element dimension during adaptive processing. These subsets are formed in an overlapped fashion along PRI dimension. Each adaptation involves $KN \times KN$ dimension linear equations which are to be solved by QR decompositions. After weight vectors are applied, Doppler processing (one M' -point FFT) is performed for a given snapshot. The weight computation steps of ESPsD are as follows: first, Doppler processing is performed separately on each element. This requires N M -point FFT's along PRI dimension. Then, a different adaptive problem is solved for each Doppler bin, utilizing all elements. M QR decompositions of size $N \times N$ are performed and data cube is accessed along element dimension, as shown in Figure 3.12. Figure 3.13 shows the computational requirements of ESPrD and ESPsD.

For the range of $16 \leq M, N \leq 128$, the estimated *flop* counts are shown in Figure 3.14, 3.15, 3.16, 3.17. It can be easily observed that the *flop* count grows exponentially as N grows. This results from the fact that performing QR decomposition dominates other operations. Thus, the computational complexities of ESPrD and ESPsD are both $O(LMN^3)$.



(a) Sample arrival sequence



(b) Input data to STAP

Figure 3.10: The input data cube and sample arrival sequence

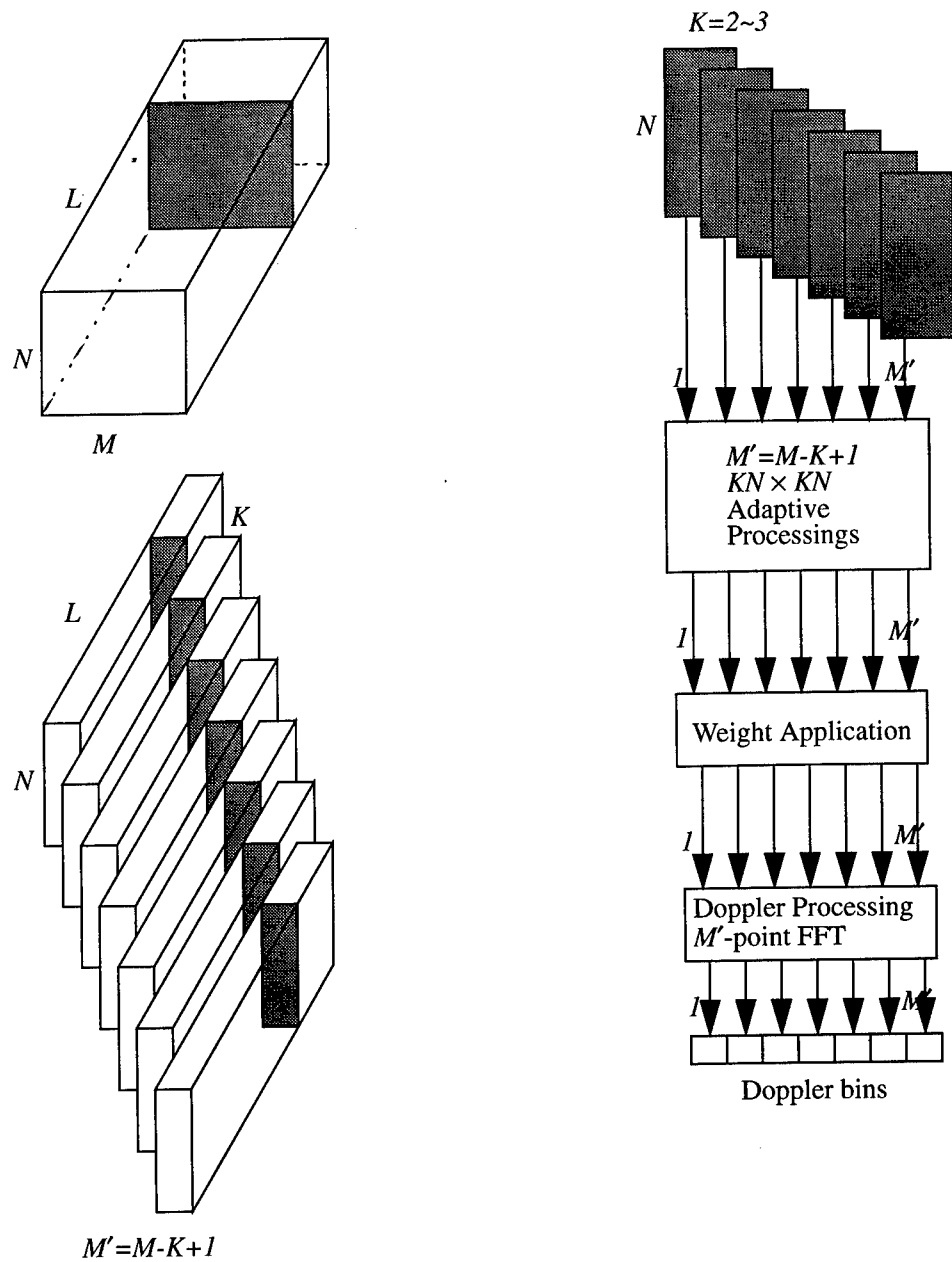


Figure 3.11: Weight Computation steps for ESPrD

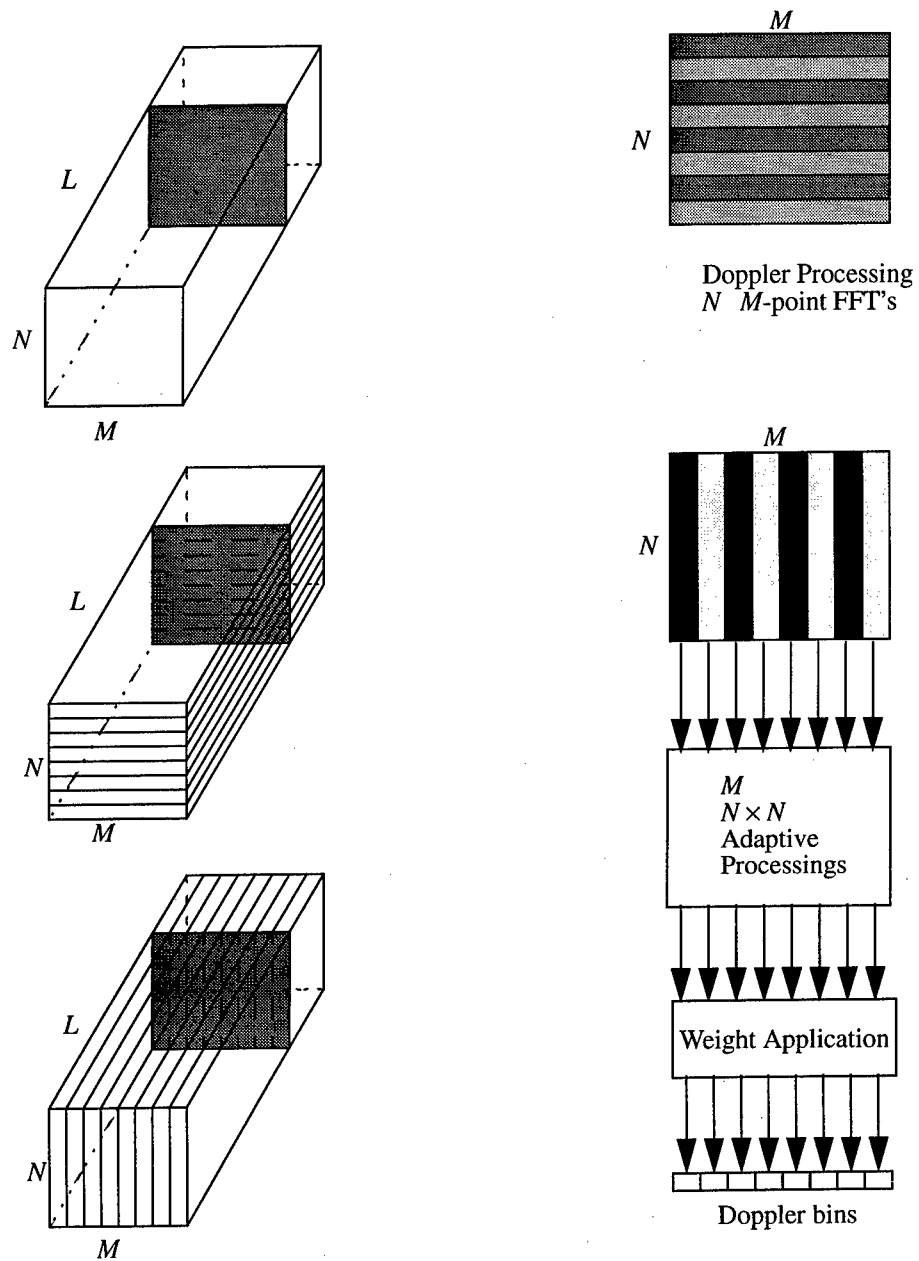


Figure 3.12: Weight Computation steps for ESPsD

ESPrD

- Adaptive Processing:
(LM' QR decompositions of size $KN \times KN$)
 $2LM'(KN)^3/3$ flops
- Doppler Processing:
($L M'$ -point FFT's)
 $5LM' \log M'$ flops

ESPsD

- Doppler Processing:
(LN M -point FFT's)
 $5LNM \log M$ flops
- Adaptive Processing:
(LM QR decompositions of size $N \times N$)
 $2LMN^3/3$ flops

Figure 3.13: Computational Complexities for ESPrD and ESPsD

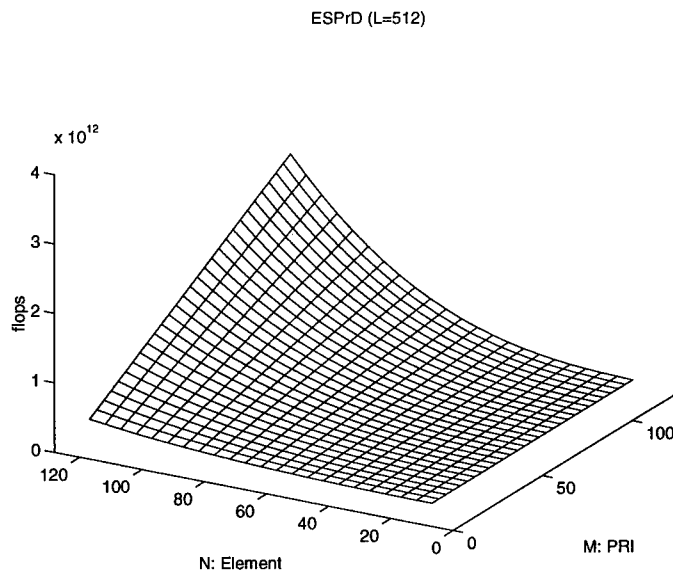


Figure 3.14: Computational Requirements of ESPrD when $L = 512$

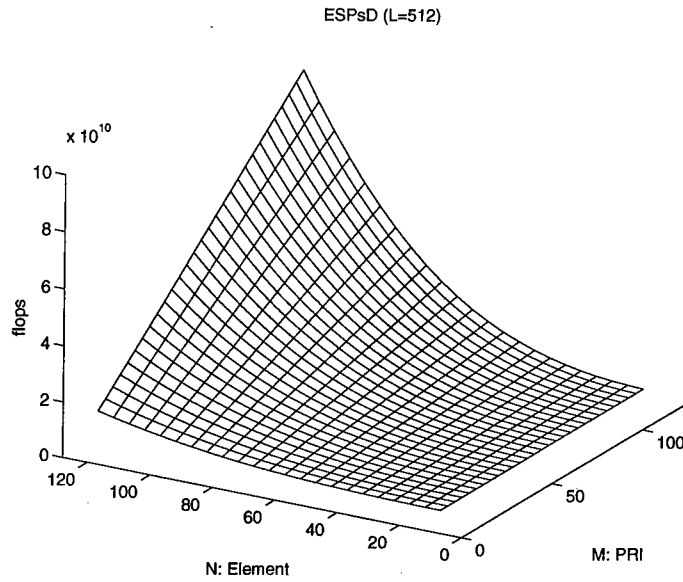


Figure 3.15: Computational Requirements of ESPsD when $L = 512$

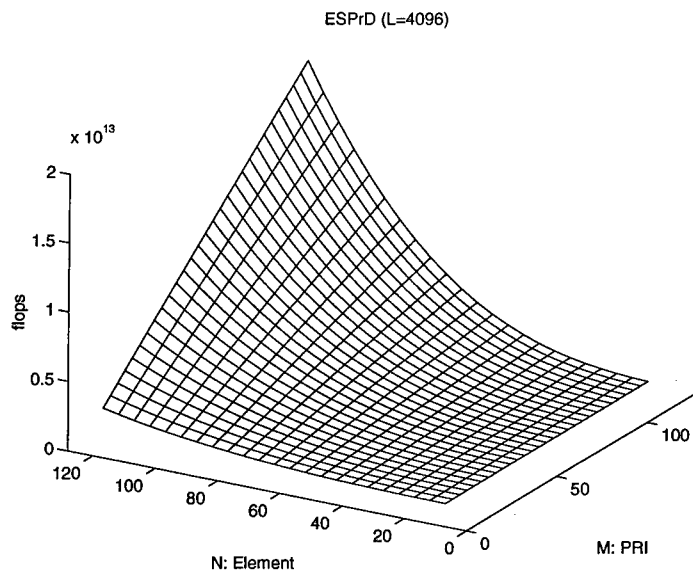


Figure 3.16: Computational Requirements of ESPrD when $L = 4096$

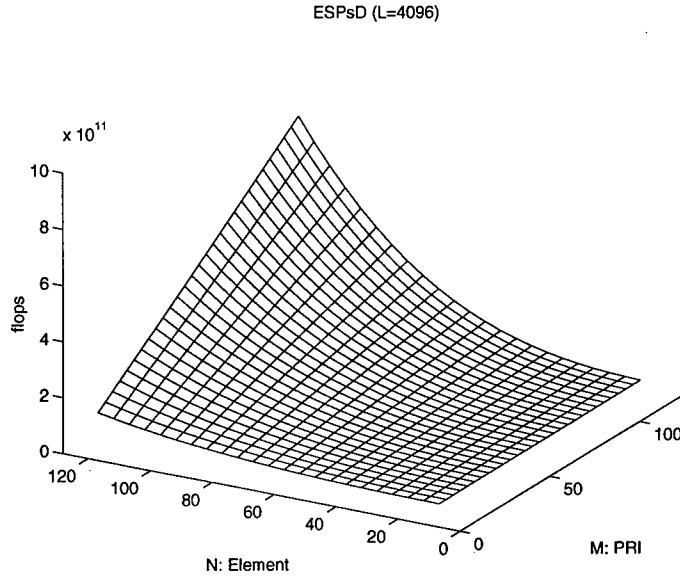


Figure 3.17: Computational Requirements of ESPsD when $L = 4096$

3.8 Scalable Portable Algorithms for ESPrD and ESPsD

In this section, we illustrate the design of scalable parallel algorithms for ESPrD and ESPsD. Let x_{nml} be the sample signal from the n -th element, m -th pulse, and at the l -th range-gate. The N antenna elements independently transmit/receive signals and compose part of the data cube. In implementing the STAP tasks on a parallel HPC platform, we assume that N I/O processors (e_0, \dots, e_{N-1}) collect the data from each element and distribute them to P computing processors (p_0, \dots, p_{P-1}). The distribution of the data onto the processors affects the communication requirements, the degree of concurrency, and the load on the processors. Our goal is to design efficient algorithms which can provide high-speed execution on available machine sizes and realistic problem sizes. For scalable performance, efficient data layout and data distribution schemes should be developed so that the parallelizing overhead is minimal; these schemes should minimize the inter-processor communication time. We parallelize weight computation steps of ESPrD and ESPsD at the task level. This means that individual tasks (FFT and QR decomposition) are performed sequentially on a single processor. However, parallelism is achieved by distributing whole tasks among the processing elements. The reason for employing task level parallelism is that the problem sizes of the tasks in ESPrD are too small to parallelize efficiently. Our previous experience [9] shows that the cumulative communication overheads incurred in parallel FFT and QR decomposition algorithms (such as ScaLAPACK routines [7]), prevents scalable performance. We choose data mapping schemes, so that each node p_i has all the data necessary for its computation.

In this data mapping scheme, the communication occurs when N I/O processing nodes distribute their data to P computing nodes. This distribution constitutes an all-to-all communication pattern. It should be noted that the number of computing nodes (P) is much larger than

the number of I/O processing nodes (N), *i.e.* $P \gg N$. This is because a large number of computing nodes are necessary to deal with the enormous computational requirements for real-time needs. We develop efficient distribution algorithms when a small set of processors distribute their data equally to a large set of processors. This happens frequently in the general real-time signal processing applications, because the number of sensors is limited while the number of processing nodes required for real-time needs is much greater. Also, when exploiting task parallelism, compute intensive tasks must be distributed among many processors to alleviate computational bottlenecks.

A simple scheduling scheme consists of P steps of communication with messages of size ML/P as shown Figure 3.18. Using the GDM model, this communication needs T_{comm_1} time, where

$$T_{comm_1} = P \cdot T_d + ML \cdot \tau_d \quad (3.12)$$

This is because at most N communications can be performed in parallel due to node contention, while there are NP communications to be performed. Thus, the cumulative start up overheads become large as P grows. Figure 3.19 shows communication scheduling which reduces cumulative start up overheads. It partitions P computing nodes into $K = \frac{P}{N}$ groups. Each group has N computing nodes. Thus, each I/O processing node can distribute $\frac{ML}{K}$ size data to first group, then second group, and so on. This distribution takes $T_d + \frac{ML}{K} \tau_d$. As soon as all processing nodes in a group receive data from I/O processing nodes, the remapping operation can be started. The timing diagram shown in Figure 3.20(a) illustrates this pipelined communication scheduling. Because the remapping within a group takes $(N - 1)\{T_d + \frac{ML}{KN} \tau_d\}$, total communication time becomes

$$T_{comm_2} = K\{T_d + \frac{ML}{K} \tau_d\} + (N - 1)\{T_d + \frac{ML}{KN} \tau_d\} \quad (3.13)$$

$$(\frac{P}{N} + N - 1)T_d + ML(1 + \frac{N - 1}{P})\tau_d \quad (3.14)$$

In ESPsD, there are many FFT computations to be performed after the first distribution. Therefore, we can overlap the remapping process with FFT computations as shown in Figure 3.20(b). Also, separate communicators for distribution and remapping processes can be used, which MPI provides to avoid unnecessary synchronization during message passing. Figure 3.21 and Figure 3.22 show the estimated communication time on SP2 and T3D for various values of L , M , and N .

3.9 Conclusion

In this work, we have developed scalable portable algorithms for partially adaptive STAP. The main contributions are in data remapping, scheduling and overlapping computation with communication. Our techniques result in uniform distribution of load on the processors as well as in minimization of the communication time. Earlier implementations are not scalable and the communication overheads dominate the overall execution time for a large number of processors. The implementations were performed using MPI and can be easily ported to other machines, including emerging HPC platforms for embedded systems applications such as the Martin Marietta Lab's High Performance Scalable Computing System [3].

We have also parallelized weight computation steps of ESPrD (Element Space Pre-Doppler) and ESPsD (Element Space Post-Doppler) STAP. We have developed efficient distribution algorithms when a small set of processors distribute their data equally to a large set of processors. We

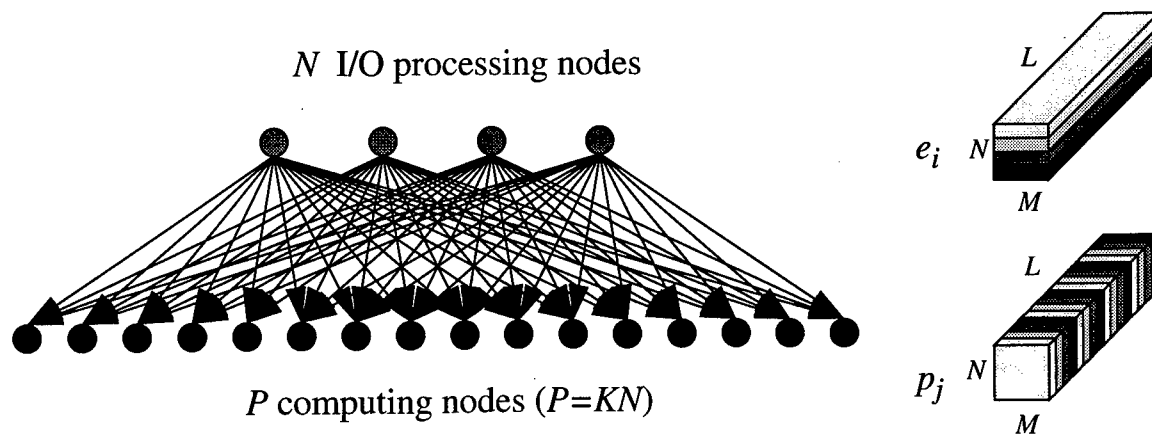


Figure 3.18: Simple Communication Scheduling

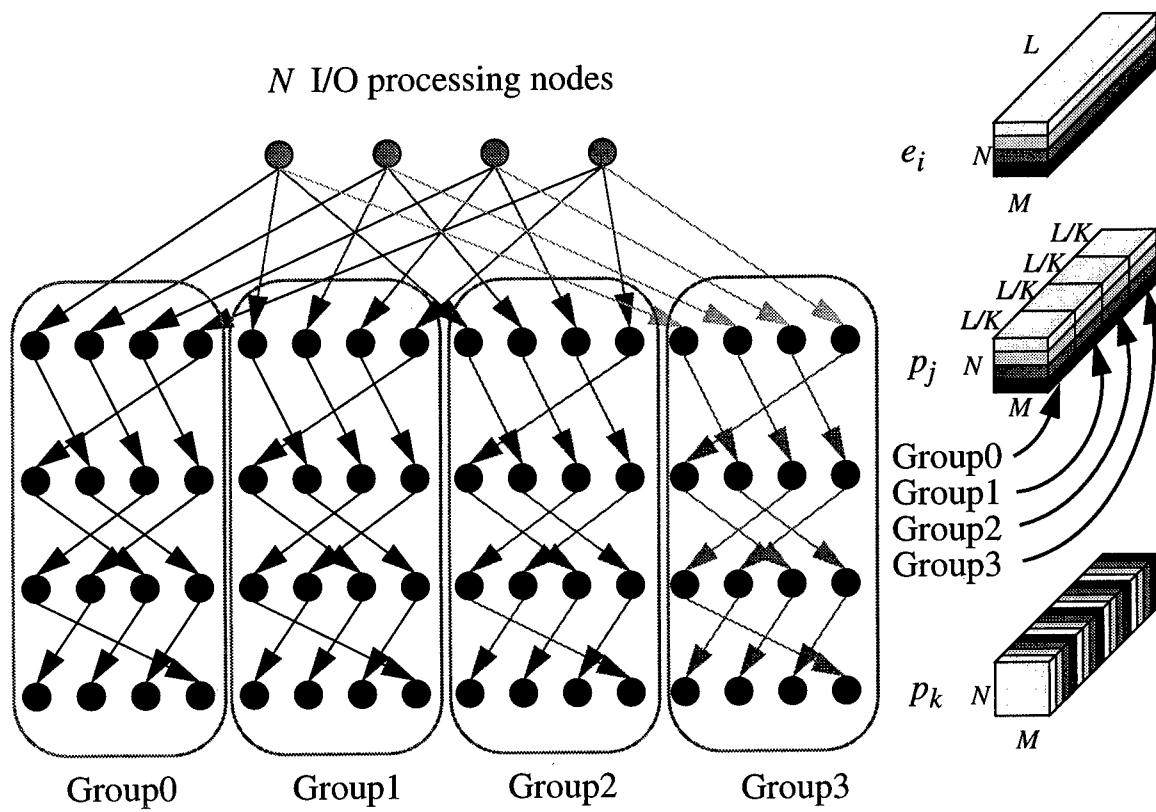
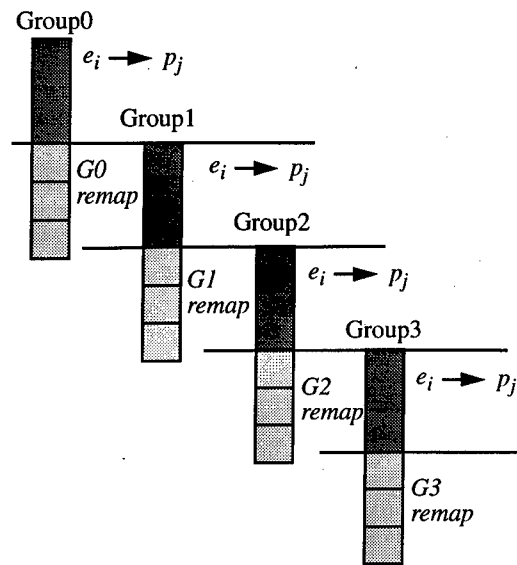
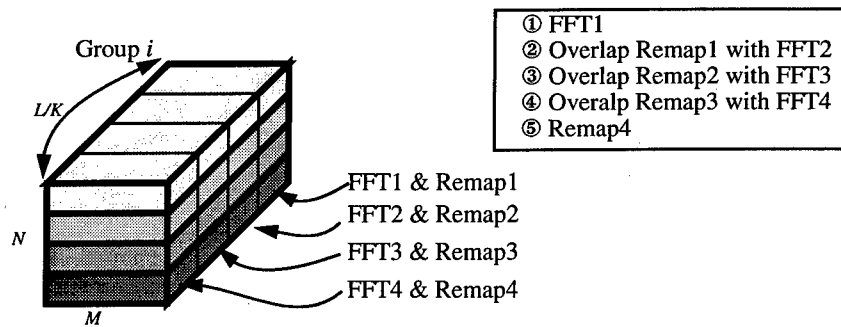


Figure 3.19: Pipelined Communication Scheduling



(a) Timing diagram



(b) Overlapping communications with computations

Figure 3.20: Timing diagram and overlapped communication with computation

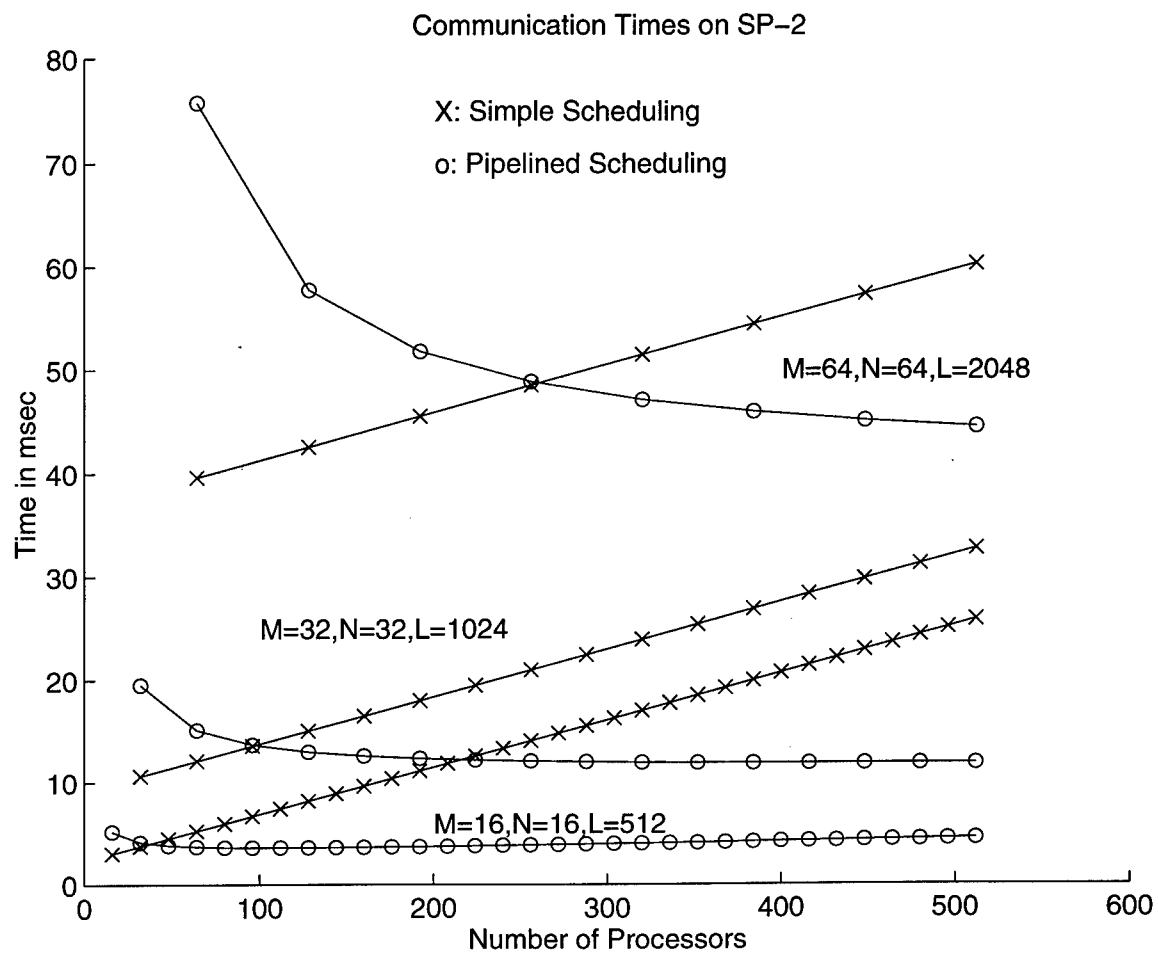


Figure 3.21: Estimated communication time on SP2 (o: Pipelined schedule, x: Simple Schedule) for $N,M,L = (64,64,2048), (32,32,1024), (16,16,512)$.

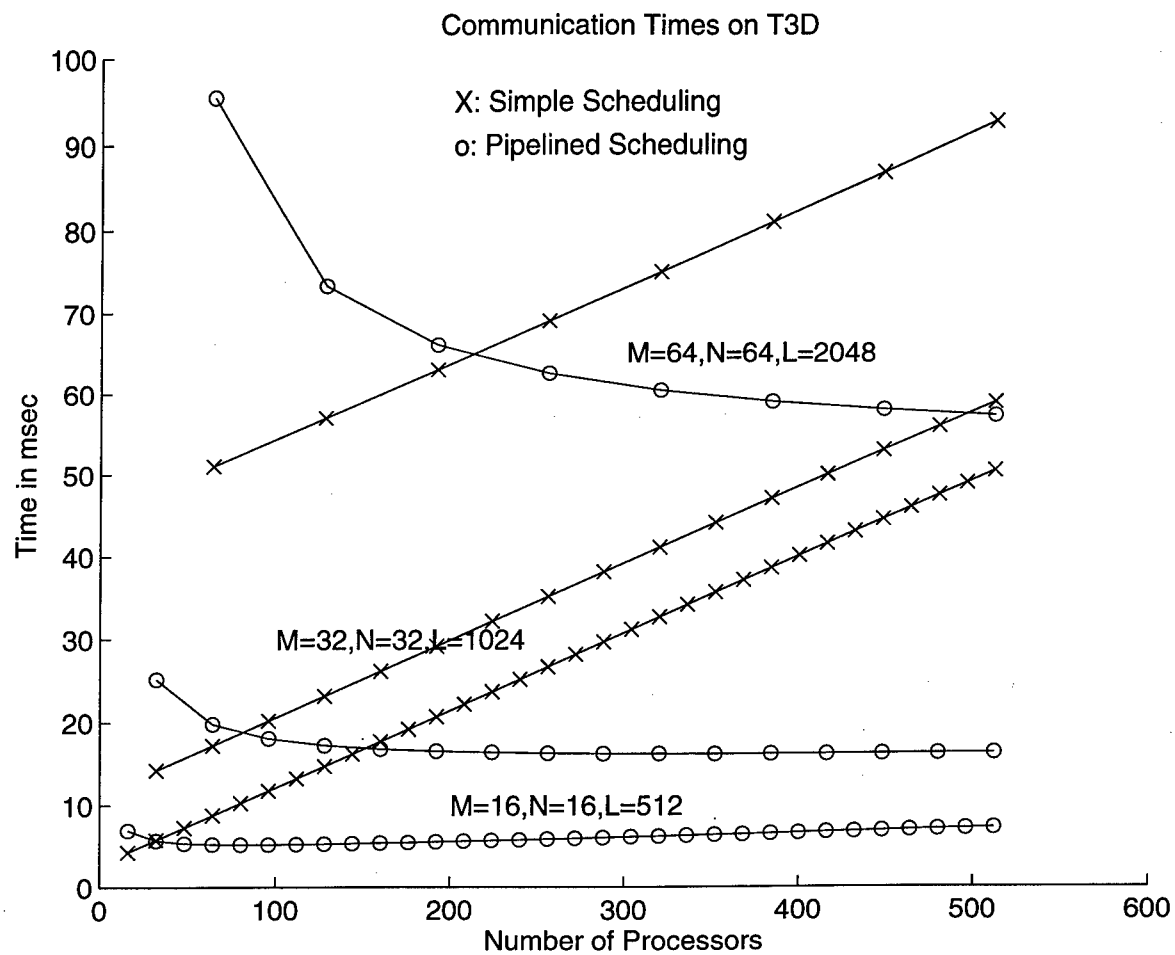


Figure 3.22: Estimated communication time on T3D (o: Pipelined schedule, x: Simple Schedule) for $N, M, L = (64, 64, 2048), (32, 32, 1024), (16, 16, 512)$.

have developed efficient distribution algorithms by using data remapping and processor groups, which will therefore reduce the effect of node contention significantly. Although we have focused on two variations of STAP, the techniques of communication scheduling and overlapping computation with communication are adequately general. They can therefore be applied to several other signal processing computations to yield efficient implementations.

Our algorithms can also be easily ported to shared memory systems. The *Prefetch* command can be used for latency hiding, thus, the computation and communication can be overlapped. It is easy to perform a similar analysis as shown in this section. For example, on shared memory machine models, one can easily perform the analysis using the Block Distributed Memory model [9].

Emerging HPC platforms provide many features for communication that can be accessed via standards such as MPI. Algorithmic techniques can exploit these to lead to scalable portable algorithms for Linear Algebra problems that offer superior performance compared with ScaLAPACK for the size of problems encountered by the signal processing and ATR communities.

Bibliography

- [1] J. Ward, "Space-Time Adaptive Processing for Airborne Radar," MIT Lincoln Lab. Technical Report 1015, Dec. 1994.
- [2] I. S. Reed, Y. L. Gau, and T. K. Truong, "A Simultaneous CFAR Detection and ML Estimation STAP Algorithm for Airborne Radar," *submitted to IEEE Trans. on AES*, 1995
- [3] R. Graybill, ARPA Embedded Systems PI meeting, Mar. 1995.
- [4] H. R. Ward, "The RAMPS PSR, A Solid State Surveillance Radar," *IEE Int. Radar Conf.*, Oct. 1987.
- [5] K. W. Przytula *et al.*, *Application Specific Array Processors*, IEEE Computer Society Press, 1990.
- [6] S. Y. Kung *et al.*, "Wavefront Array Processors – Concept to Implementation," *Computer Society of IEEE*, Vol. 20, No. 7, pp. 18-33, Jul. 1987.
- [7] J. Choi, J. J. Dongarra, L. S. Ostrouchov, A. P. Petitet, D. W. Walker, and R. C. Whaley, "The Design and Implementation of the SCALAPACK LU, QR, and Cholesky Factorization Routines," Oak Ridge National Lab. Technical Report ORNL/TM-12470, Dec. 1994.
- [8] V. K. Prasanna and C. Wang, "Scalable Parallel Implementations of Perceptual Grouping on Connection Machine CM-5," *Proceedings of International Conference on Pattern Recognition*, Oct. 1994.
- [9] J. F. Já Já and K. W. Ryu, "The Block Distributed Memory Model for shared memory multiprocessors," in *International Parallel Processing Symposium*, pp. 752 -756, Apr. 1994.
- [10] Y. W. Lim and V. K. Prasanna, "Adaptive Beamforming on distributed memory machines," in preparation.
- [11] S. J. Olszanskyj, J. M. Lebak, and A. W. Bojanczyk, "Parallel Algorithms for Space-Time Adaptive Processing," *International Parallel Processing Symposium '95*, pp. 77-81, Apr. 1995.
- [12] D. P. O'Leary and P. Whitman, "Parallel QR factorization by Householder and modified Gram-Schmidt algorithms," *Parallel Computing*, Vol. 16, pp. 99-112, 1990.
- [13] J. F. Já Já, *An Introduction to Parallel Algorithms*, Addison-Wesley, 1992.

- [14] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, 1983.
- [15] V. Kumar *et al.*, *Introduction to Parallel Computing: Design and Analysis of Algorithms*, The Benjamin/Cummings Publishing Company, Inc., 1994.
- [16] N. J. Boden, D. Cohen, R. E. Felderman, A. E. Kulawik, C. L. Seitz, J. N. Seizovic, and W.-K. Su, "Myrinet - A Gigabit-per-Second Local-Area Network," *IEEE Micro*, Feb. 1995.
- [17] Message Passing Interface Forum. "MPI: A Message-Passing Interface Standard," *Technical Report CS-94-230, University of Tennessee, Knoxville, TN*, May 5 1994.
- [18] J. Ward, "Space-Time Adaptive Processing for Airborne Radar," *MIT Lincoln Lab., Technical Report 1015*, Dec 1994.

DISTRIBUTION LIST

addresses	number of copies
ATTN: WILLIAM L. MELVIN, CAPT, USAF ROME LABORATORY/OCSS 26 ELECTRONIC PKY ROME, NY 13441-4514	5
UNIVERSITY OF SOUTHERN CALIFORNIA ATTN: IRVING REED DEPT OF ELECTRICAL ENGINEERING UNIVERSITY PARK LOS ANGELES, CA 90089-2564	3
ROME LABORATORY/SUL TECHNICAL LIBRARY 26 ELECTRONIC PKY ROME NY 13441-4514	1
ATTENTION: DTIC-OCC DEFENSE TECHNICAL INFO CENTER 8725 JOHN J. KINGMAN ROAD, STE 0944 FT. BELVOIR, VA 22060-6218	2
ADVANCED RESEARCH PROJECTS AGENCY 3701 NORTH FAIRFAX DRIVE ARLINGTON VA 22203-1714	1
ATTN: RAYMOND TADROS GIDEP P.O. BOX 8000 CORONA CA 91718-8000	1
ATTN: WALTER HARTMAN WRIGHT LABORATORY/AAM, BLDG. 620 2241 AVIONICS CIRCLE, RM N3-F10 WRIGHT-PATTERSON AFB OH 45433-7333	1
AFIT ACADEMIC LIBRARY/LDEE 2950 P STREET AREA B, BLDG 642 WRIGHT-PATTERSON AFB OH 45433-7765	1

ATTN: R.L. DENISON
WRIGHT LABORATORY/MLPO, BLDG. 651
3005 P STREET, STE 6
WRIGHT-PATTERSON AFB OH 45433-7707

1

WRIGHT LABORATORY/FIVS/SURVIAC
2130 EIGHTH STREET, BLDG 45, STE 1
WRIGHT-PATTERSON AFB OH 45433-7542

1

ATTN: GILBERT G. KUPERMAN
AL/CFHI, BLDG. 248
2255 H STREET
WRIGHT-PATTERSON AFB OH 45433-7022

1

AUL/LSAD
600 CHENNAULT CIRCLE, BLDG. 1405
MAXWELL AFB AL 36112-6424

1

US ARMY STRATEGIC DEFENSE COMMAND
CSSD-IM-PA
P.O. BOX 1500
HUNTSVILLE AL 35807-3801

1

NAVAL AIR WARFARE CENTER
6000 E. 21ST STREET
INDIANAPOLIS IN 46219-2189

1

COMMANDING OFFICER
NCCOSC RDT&E DIVISION
ATTN: TECHNICAL LIBRARY, CODE 0274
53560 HULL STREET
SAN DIEGO CA 92152-5001

1

COMMANDER, TECHNICAL LIBRARY
474700D/C0223
NAVAIRWARCENWPNDIV
1 ADMINISTRATION CIRCLE
CHINA LAKE CA 93555-6001

1

SPACE & NAVAL WARFARE SYSTEMS
COMMAND (PMW 178-1)
2451 CRYSTAL DRIVE
ARLINGTON VA 22245-5200

2

SPACE & NAVAL WARFARE SYSTEMS
COMMAND, EXECUTIVE DIRECTOR (PD13A)
ATTN: MR. CARL ANDRIANI
2451 CRYSTAL DRIVE
ARLINGTON VA 22245-5200

1

COMMANDER, SPACE & NAVAL WARFARE
SYSTEMS COMMAND (CODE 32)
2451 CRYSTAL DRIVE
ARLINGTON VA 22245-5200

1

CDR, US ARMY MISSILE COMMAND
RSIC, BLDG. 4484
AMSMI-RD-CS-R, DOCS
REDSTONE ARSENAL AL 35898-5241

2

ADVISORY GROUP ON ELECTRON DEVICES
SUITE 500
1745 JEFFERSON DAVIS HIGHWAY
ARLINGTON VA 22202

1

REPORT COLLECTION, CIC-14
MS P364
LOS ALAMOS NATIONAL LABORATORY
LOS ALAMOS NM 87545

1

AEDC LIBRARY
TECHNICAL REPORTS FILE
100 KINDEL DRIVE, SUITE C211
ARNOLD AFB TN 37389-3211

1

AFIWC/MSO
102 HALL BLVD, STE 315
SAN ANTONIO TX 78243-7016

1

NSA/CSS
K1
FT MEADE MD 20755-6000

1

PHILLIPS LABORATORY
PL/TL (LIBRARY)
5 WRIGHT STREET
HANSCOM AFB MA 01731-3004

1

THE MITRE CORPORATION
ATTN: E. LADURE
D460
202 BURLINGTON RD
BEDFORD MA 01732

1

OUSDC(P)/DTSA/DUTD
ATTN: PATRICK G. SULLIVAN, JR.
400 ARMY NAVY DRIVE
SUITE 300
ARLINGTON VA 22202

2

MISSION OF ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

- a. Conducts vigorous research, development and test programs in all applicable technologies;
- b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;
- c. Provides a full range of technical support to Air Force Material Command product centers and other Air Force organizations;
- d. Promotes transfer of technology to the private sector;
- e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.